CY7C09449PVA-AC

128 Kb Dual-Port SRAM with PCI Bus Controller (PCI-DP)

Features

- 128 Kb of dual-ported shared memory
- Master and target PCI Specification 2.2 compliant interface
- Embedded host bridge capability
- Direct interface to many microprocessors
- I_2O message transport unit; includes four 32-bit, 32 entry FIFO
- Local bus clock rates up to 50 MHz
- Single 3.3V power supply including compatibility with 3V and 5V PCI bus signaling
- 160-pin thin plastic quad flat package

Introduction

The CY7C09449PVA is one of the PCI interface controllers in the Cypress Semiconductor PCI-DP™ family. The CY7C09449PVA provides a PCI master and target interface with direct connections to many popular microprocessors. It provides 128 Kb of dual-port SRAM that is used as shared memory between the local microprocessor and the PCI bus. An I_2O message unit, complete with message queues and interrupt capability, is also provided. The CY7C09449PVA allows the designer to interface an application to the PCI bus in a straightforward and inexpensive way.

Functional Overview

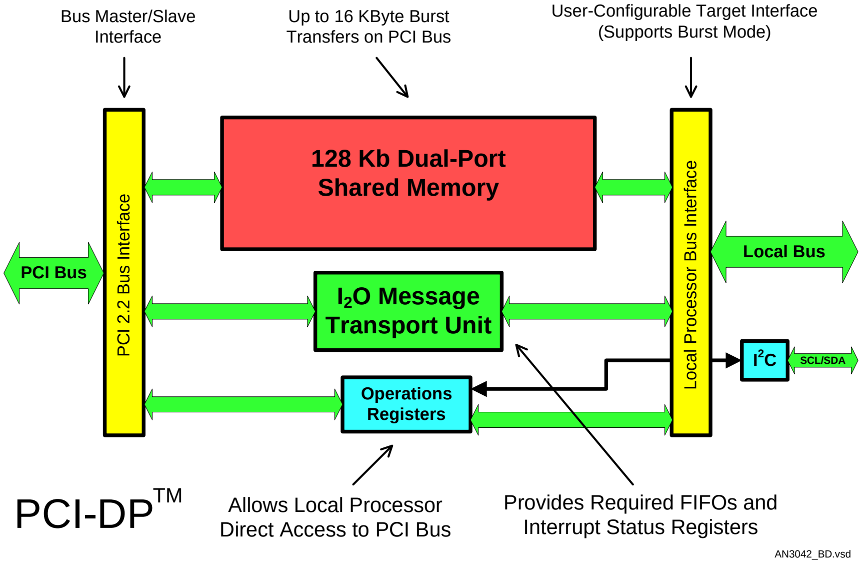
The CY7C09449PVA is composed of several shared resources that allow effective data movement between the local bus and the PCI bus.

A primary resource within the CY7C09449PVA is its 128 Kb of dual-port memory. This memory is interfaced to both the PCI bus and a local microprocessor bus. This shared memory is accessed as a target from both buses at the same time for inter process communication. The CY7C09449PVA is directed from both the local and PCI bus to become a PCI bus master and move data into or out of the internal shared memory as a direct memory access (DMA). The CY7C09449PVA can DMA across the PCI bus any number of 32-bit double words (DWORD), up to 16K bytes. It uses the full bursting capabilities of the PCI bus for maximum efficiency and transfers data over the full 32-bit PCI address space.

The CY7C09449PVA implements optional requirements of the PCI specification by selecting the optimum PCI command for each transaction it masters to the PCI bus. This maximizes the overall efficiency of the system platform. PCI bridging functions (PCI-to-PCI and Host-to-PCI bridges) use the commands to enhance prefetch and cache coherency operations. The CY7C09449PVA requests and gains access to the PCI bus as any master. It does not include a PCI bus arbitration function. Standard PC PCI buses include this function; embedded systems may need to implement this function.

The CY7C09449PVA provides a direct access mechanism from the local bus to the PCI bus. With it, the local processor directs the CY7C09449PVA to run a PCI bus master cycle of any kind to any address. This means that the CY7C09449PVA runs PCI configuration cycles as a host bridge.

Logic Block Diagram



Bus Master/Slave Interface
Up to 16 KByte Burst Transfers on PCI Bus
User-Configurable Target Interface (Supports Burst Mode)
128 Kb Dual-Port Shared Memory
PCI Bus
PCI 2.2 Bus Interface
Local Bus
Local Processor Bus Interface
I2O Message Transport Unit
I2C
SCL/SDA
Operations Registers
PCI-DP™
Allows Local Processor Direct Access to PCI Bus
Provides Required FIFOs and Interrupt Status Registers
AN3042_BD.vsd

Four First In First Out (FIFO) storage elements provide another resource to the user. These are accessible from either the PCI bus or the local bus. When the I_2O messaging unit functionality of the CY7C09449PVA is used, the four FIFOs become part of the I_2O messaging unit. The I_2O messaging unit consists of the four FIFOs and the I_2O system interrupt registers. The shared memory of the CY7C09449PVA is used to store I_2O message frame buffers; most of the shared memory is still available for general use. Efficient I_2O messaging is realized when the local processor uses the CY7C09449PVA direct access mechanism. It is used to retrieve and post I_2O message pointers to other I_2O agents. Data transfer of the messages themselves is made very efficient using the CY7C09449PVA PCI DMA controller to burst the message frames to other I_2O agents.

Interprocess communication is supported by two resources of the CY7C09449PVA: the mailbox registers and arbitration flags. When writing to the mailbox registers, a method is available for the local processor to pass data while causing an interrupt to the host, and vice versa. This is enabled by the interrupt mask located in the CY7C09449PVA Operations Registers. The arbitration flags are four pairs of bits that are used to manage resource allocation and sharing between software and system processes.

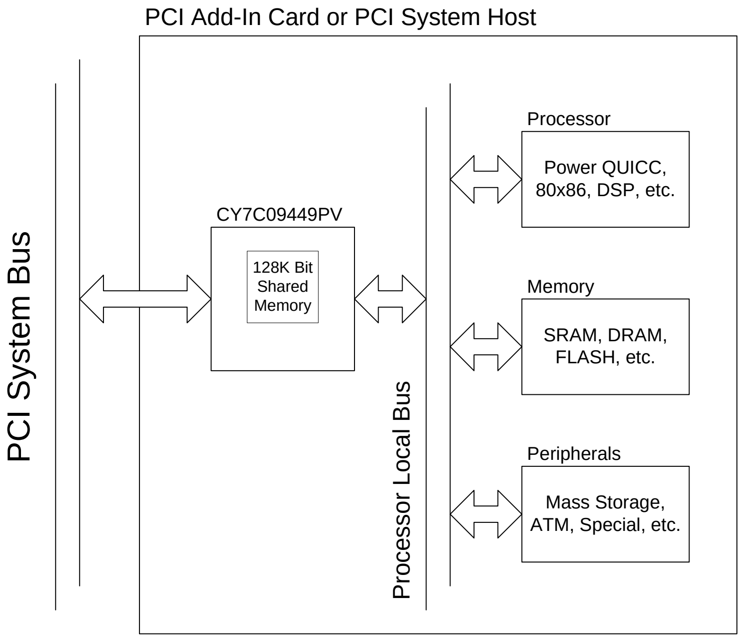
The CY7C09449PVA includes an interrupt controller. There are separate interrupt masks and command and status registers for both PCI bus and local bus. The interrupt sources are DMA completion, mailbox, FIFO not empty (also for I_2O), FIFO overflow, PCI master abort, PCI target abort, and an external interrupt input pin. This interrupt controller is used to signal interrupts onto the PCI bus and the local bus. The CY7C09449PVA interrupt controller does not perform the interrupt controller function for the PCI bus system. Standard PC PCI systems include this function; embedded systems may need to implement this function.

An I^2C compatible serial interface is provided to allow the use of a serial EEPROM for non-volatile storage of CY7C09449PVA initialization parameters. The parameters are PCI configuration and local bus settings. The CY7C09449PVA optionally accesses the EEPROM after reset and downloads initialization information before responding to PCI or local bus transactions. A wide variety of I^2C compatible serial components are available to the local and host processor when connected through this interface.

The CY7C09449PVA local bus is a flexible, configurable interface that is designed to readily connect to many industry standard microprocessors. In most cases, no external interface logic (“glue”) is needed.

The following block diagram illustrates a generic application for the CY7C09449PVA.

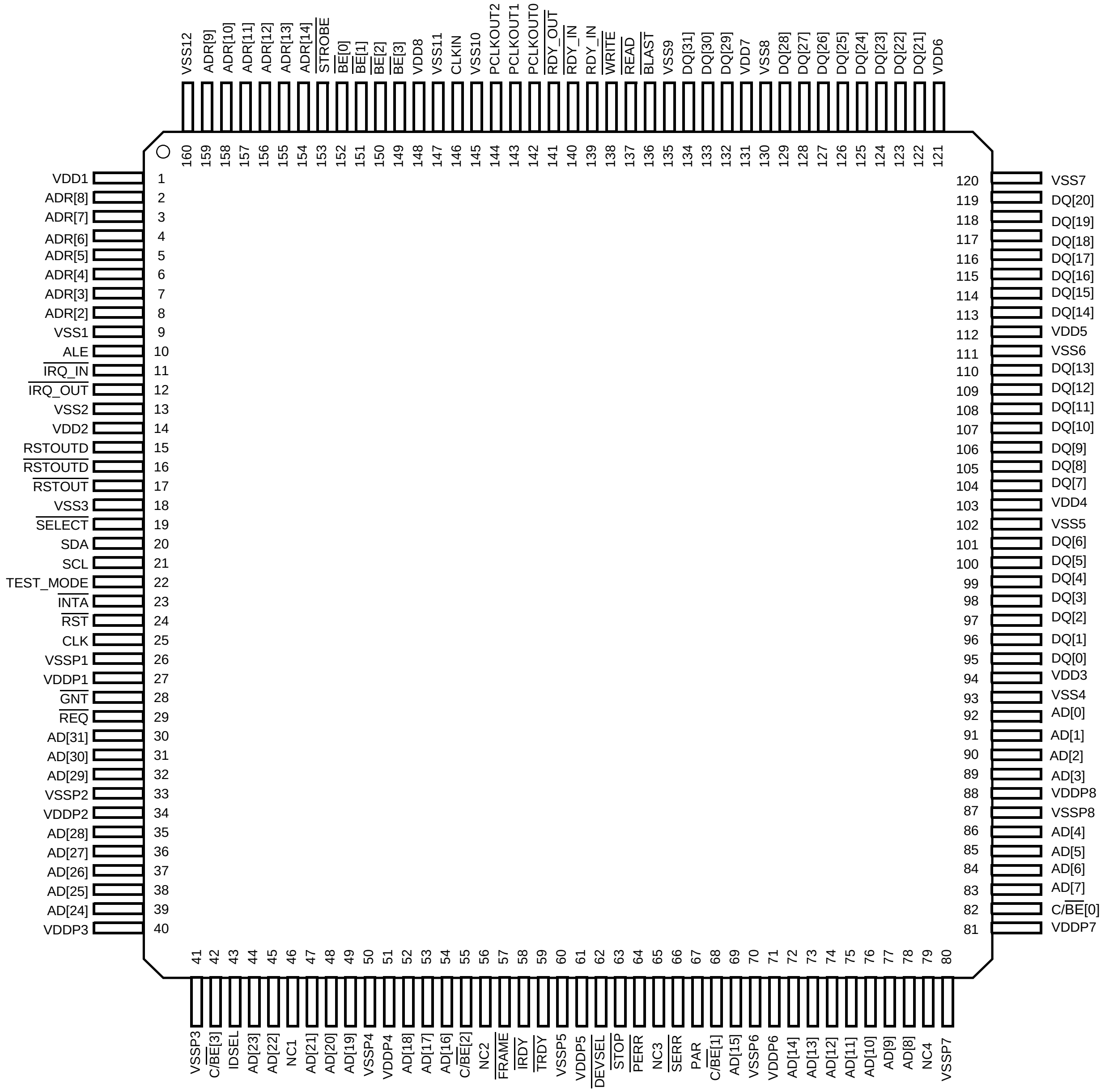
Figure 1. Generic Application Block Diagram



3042APP.VSD DB 6/02/00

Pinouts

Figure 2. Pin Diagram: 160-Pb TQFP (A160) for CY7C09449PVA Top View



Pin Description

Table 1. Pin Type for CY7C09449PVA

Pin	Description
in	Input is a standard input only signal.
out	Standard output driver.
t/s	Tri-state is an output or bidirectional signal.
s/t/s	Sustained tri-state is an active LOW, tri-state capable signal driven by only one bus agent at a time. When ownership is passed to another agent, the signal is driven HIGH for one clock, and then tri-stated for an additional clock before being driven by the new owner.
o/d	Open Drain signals allow multiple devices to share the pin as a wired OR.
o/c	Open Collector signals allow multiple devices to share the pin as a wired OR.

Table 2. PCI Bus Signals

Signal Name	Type	Description
CLK	in	**Clock.** This is the PCI bus clock and the timing reference for all PCI bus transactions. The CY7C09449PVA operates with a 33 MHz PCI bus interface.
$\overline{\text{RST}}$	in	**RESET.** This signal is the PCI bus reset. It is one of the few PCI signals that may be asserted or deasserted asynchronously to the PCI bus clock (CLK).
AD[31:0]	t/s	**Address and Data.** These signals represent the PCI bus address and data signals multiplexed on the same PCI pins. Information on these pins is identified as an address during the clock cycle in which the signal FRAME is first asserted. This is termed the “address phase” of a bus transaction. Information on these pins represents valid read or write data when both IRDY and TRDY are asserted, based on the current cycle type as defined on the C/BE lines during the address phase. This condition is termed the “data phase” of a bus transaction.
C/$\overline{\text{BE}}$[3:0]	t/s	**Command and Byte Enables.** These pins are used with the AD[31:0] pins. During the address phase of a bus operation, they identify the bus command performed. During the data phase of a bus operation they identify which bytes are involved.
PAR	t/s	**Parity.** This PCI bus pin represents the even parity across the A/D[31:00] and C/$\overline{\text{BE}}$[3:0] pins (36 pins) and is generated with a one clock delay.
$\overline{\text{FRAME}}$	s/t/s	**Cycle Frame.** This PCI bus pin is asserted by the current bus master to signify the beginning of a bus transaction. Data transfers may continue in burst mode while FRAME remains asserted. When FRAME is deasserted it indicates that the transaction is in the final data phase.
$\overline{\text{IRDY}}$	s/t/s	**Initiator Ready.** This signal is driven by the current bus master (initiator) and asserted to indicate its ability to complete the current data phase. Data is transferred when both IRDY and TRDY are asserted, otherwise wait cycles occur.
$\overline{\text{TRDY}}$	s/t/s	**Target Ready.** This signal is driven by the selected bus target and asserted when that target is ready to complete the current data phase. Data is transferred when both IRDY and TRDY are asserted, otherwise wait cycles occur.
$\overline{\text{STOP}}$	s/t/s	**Stop.** The $\overline{\text{STOP}}$ signal is driven by the selected bus target and is asserted when it wishes to cease the current data transaction.
IDSEL	in	**Initialization Device Select.** This signal is used to gain access to the PCI configuration register space of a given PCI Bus agent.
$\overline{\text{DEVSEL}}$	s/t/s	**Device Select.** The $\overline{\text{DEVSEL}}$ signal is driven and asserted by the currently selected PCI bus target based on the current address and that target’s assigned address range. Bus masters examine this signal to determine whether the desired device is present.

Table 2. PCI Bus Signals (continued)

Signal Name	Type	Description
$\overline{\text{REQ}}$	t/s	**Request.** This signal indicates to the bus arbiter that this device wishes to use the bus. It is a point to point signal, which is driven whenever RST is not asserted.
$\overline{\text{GNT}}$	t/s	**Grant.** This point to point signal indicates that the bus is granted to the requester. It is driven whenever RST is not asserted and is ignored during the assertion of RST.
$\overline{\text{PERR}}$	s/t/s	**Parity Error.** This signal indicates that a parity error has occurred. It is driven by the target or master that receives data at the clock after the PAR signal becomes valid.
$\overline{\text{SERR}}$	o/d	**System Error.** This open drain signal is driven by any device that detects odd parity during an address phase.
$\overline{\text{INTA}}$	o/d	**Interrupt A.** This signal is asserted when interrupt servicing of the CY7C09449PVA device is required. The INTA pin is a shared PCI bus signal and uses an open drain element to allow a wired OR.

Table 3. Local Bus Interface Signals

Signal Name	Type	Description
ADR[14:2]	in	**Address.** These signals identify the local memory location. When the local processor outputs multiplexed address and data, those lines are tied to both the DQ[14:2] and ADR[14:2].
$\overline{\text{BE}}$[3:0]	in	**Byte Enables.** The byte enable inputs identify the specific bytes involved in an access. The pins may be configured as byte lane enables directly, or used as size and encoded byte lane enables when interfacing to certain Motorola processors. See the section Local Bus on page 11 for definition.
DQ[31:0]	t/s	**Data.** CY7C09449PVA data input and output are provided on these bidirectional pins. This bus remains in high impedance during power up and active Reset (RST) and only drives during read transactions.
$\overline{\text{SELECT}}$	in	**Chip Select.** This signal must be asserted for the full duration of any access. The polarity is programmable; the default is active LOW.
ALE	in	**Address Latch Enable.** The local address provided on ADR[14:2] is latched on the trailing edge (from active to inactive) of this signal. The polarity is programmable; the default is active HIGH.
$\overline{\text{STROBE}}$	in	**Address Strobe.** The assertion of this signal begins a memory access and indicates that a valid address is latched through ALE or is provided at the pins (if ALE is not used and is tied active). The address is provided on the ADR[14:2] pins (during non multiplexed mode), or on the DQ[14:2] (during multiplexed mode). Outside the address phase, the level of STROBE is don't care.The polarity is programmable; the default is active LOW.
$\overline{\text{WRITE}}$ READ	in	**Write and Read Signals.** These signals control the transfer of data to and from the local data bus. WRITE and READ are sampled in the address phase and are don't cares during the remainder of the bus transaction. The polarity and function of these signals is programmable so that they are interfaced to processors that support WR/RD or RD/WR and separate RD/RD and WR/WR signals.
$\overline{\text{BLAST}}$	in	**Burst Last.** The signal indicates the end of a burst transfer. This signal has two modes. It is active during the burst and inactive when the burst is over, or it is active during the last data phase of the burst. The polarity is programmable; the default is active LOW.
$\overline{\text{RDY\_IN}}$ RDY_IN	in	**Ready In.** The assertion of these signals indicates that the local processor is prepared to complete the current data transaction.
RDY_OUT	out or t/s	**Ready Out.** When this signal is asserted it indicates that the CY7C09449PVA is ready to complete the current access. The polarity is programmable; the default is active LOW. This signal is also programmable to tri-state when inactive; the default is to tri-state when inactive.

Table 4. Local System Signals

Signal Name	Type	Description
PCLKOUT2 PCLKOUT1 PCLKOUT0	out	**Clock Outputs.** These pins provide three buffered copies of the PCI bus clock.
CLKIN	in	**Clock In.** This pin provides the timing reference for local bus signals. The CLKIN pin is driven by an external clock. One of the buffered copies of the PCI clock, PCLKOUT[2:0], is also used as input to CLKIN. This clock must be toggling for proper startup operation of the CY7C09449PVA and for PCI access to resources other than the dual-ported shared memory.
$\overline{\text{RSTOUT}}$	out	**Reset Out.** This pin provides a buffered version of the PCI bus signal, $\overline{\text{RST}}$.
$\overline{\text{RSTOUTD}}$ RSTOUTD	out	**Reset Out Delayed.** These pins are similar to the $\overline{\text{RSTOUT}}$ pin described earlier, however $\overline{\text{RSTOUTD}}$ and RSTOUTD remain asserted until released by the host interface through software control. This allows the CY7C09449PVA to hold the local processor in reset until the host processor is ready to release it.
$\overline{\text{IRQ\_OUT}}$	t/s	**Interrupt Request Out.** This signal is used to trigger an interrupt on the local microprocessor. A variety of host triggered events are used to cause the assertion of this interrupt request output. This signal is masked using the Local Interrupt Control and Status Register. When in the inactive state, this signal is tri-stated. The polarity is programmable; the default is active LOW.
$\overline{\text{IRQ\_IN}}$	in	**Interrupt Request In.** This signal, when asserted, results in the CY7C09449PVA driving the PCI bus INTA signal and therefore cause an interrupt of the host system. This signal may be masked using the Host Interrupt Control and Status Register.
TEST_MODE	in	**Test Mode.** When HIGH, this pin puts the CY7C09449PVA into a factory test mode. When HIGH and READ is LOW, all outputs are set to high impedance except RDY_OUT, which continues to drive if Operations Register LBUSCFG bit 16 is ‘0’. This is the only test mode available to the user. The user must drive this signal LOW if unused.

Table 5. Local Configuration Signals

Signal Name	Type	Description
SCL	o/c	**Serial Clock.** This pin is the clock output used with an external I^2C compatible serial memory device. A pull up resistor is required.
SDA	o/c	**Serial Data.** This pin is the bidirectional data pin used with an external I^2C compatible serial memory device. A pull up resistor is required.

Table 6. Power Pins

Signal Name	Type	Description
VSSP1–VSSP8, VSS1–VSS12	GND	**Ground.** These pins are ground pins (0V).
VDDP1–VDDP8, VDD1–VDD8	POWER	**Power.** These pins provide power, nominally 3.3V.

Table 7. Other Pins

Signal Name	Type	Description
NC1–NC4	NC	**No Connect.** These pins are not to be used; leave unconnected.

Signal Terminations

PCI bus signals are terminated according to the *PCI 2.2 Specification*. Generally, termination is provided by the PCI system. If the CY7C09449PVA is used as a PCI add in card or other device as part of a PCI bus, no termination must be used. For embedded systems, terminations are part of the system design; they are not particular to the CY7C09449PVA. Any PCI system must include a single pull up on each PCI bus control signal used. These signals are FRAME, TRDY, IRDY, DEVSEL, STOP, SERR, PERR, LOCK, INTA, INTB, INTC, INTD, REQ64, and ACK64. Refer to Section 4.3, System (Motherboard) Specification, of the *PCI 2.2 Specification* for detailed requirements.

All local system and local bus interface input signals must be driven at all times. If they are unused inputs, then they are driven either HIGH or LOW (pull up or pull down, V_{DD} or Ground).

SCL and SDA must have a pull up in the range of 2.2 kΩ to 10 kΩ to V_{DD}. These pull ups are required whether the signals are used or not.

Pin List

Pin Name	Number	Pin Name	Number	Pin Name	Number	Pin Name	Number
VDD1	1	VSSP3	41	VDDP7	81	VDD6	121
ADR[8]	2	C/$\overline{\text{BE}}$[3]	42	C/$\overline{\text{BE}}$[0]	82	DQ[21]	122
ADR[7]	3	IDSEL	43	AD[7]	83	DQ[22]	123
ADR[6]	4	AD[23]	44	AD[6]	84	DQ[23]	124
ADR[5]	5	AD[22]	45	AD[5]	85	DQ[24]	125
ADR[4]	6	NC1	46	AD[4]	86	DQ[25]	126
ADR[3]	7	AD[21]	47	VSSP8	87	DQ[26]	127
ADR[2]	8	AD[20]	48	VDDP8	88	DQ[27]	128
VSS1	9	AD[19]	49	AD[3]	89	DQ[28]	129
ALE	10	VSSP4	50	AD[2]	90	VSS8	130
$\overline{\text{IRQ\_IN}}$	11	VDDP4	51	AD[1]	91	VDD7	131
$\overline{\text{IRQ\_OUT}}$	12	AD[18]	52	AD[0]	92	DQ[29]	132
VSS2	13	AD[17]	53	VSS4	93	DQ[30]	133
VDD2	14	AD[16]	54	VDD3	94	DQ[31]	134
RSTOUTD	15	C/$\overline{\text{BE}}$[2]	55	DQ[0]	95	VSS9	135
$\overline{\text{RSTOUTD}}$	16	NC2	56	DQ[1]	96	$\overline{\text{BLAST}}$	136
$\overline{\text{RSTOUT}}$	17	$\overline{\text{FRAME}}$	57	DQ[2]	97	$\overline{\text{READ}}$	137
VSS3	18	$\overline{\text{IRDY}}$	58	DQ[3]	98	$\overline{\text{WRITE}}$	138
$\overline{\text{SELECT}}$	19	$\overline{\text{TRDY}}$	59	DQ[4]	99	RDY_IN	139
SDA	20	VSSP5	60	DQ[5]	100	$\overline{\text{RDY\_IN}}$	140
SCL	21	VDDP5	61	DQ[6]	101	$\overline{\text{RDY\_OUT}}$	141
TEST_MODE	22	$\overline{\text{DEVSEL}}$	62	VSS5	102	PCLKOUT0	142
$\overline{\text{INTA}}$	23	$\overline{\text{STOP}}$	63	VDD4	103	PCLKOUT1	143
$\overline{\text{RST}}$	24	$\overline{\text{PERR}}$	64	DQ[7]	104	PCLKOUT2	144
CLK	25	NC3	65	DQ[8]	105	VSS10	145
VSSP1	26	$\overline{\text{SERR}}$	66	DQ[9]	106	CLKIN	146
VDDP1	27	PAR	67	DQ[10]	107	VSS11	147
$\overline{\text{GNT}}$	28	C/$\overline{\text{BE}}$[1]	68	DQ[11]	108	VDD8	148
$\overline{\text{REQ}}$	29	AD[15]	69	DQ[12]	109	$\overline{\text{BE}}$[3]	149
AD[31]	30	VSSP6	70	DQ[13]	110	$\overline{\text{BE}}$[2]	150
AD[30]	31	VDDP6	71	VSS6	111	$\overline{\text{BE}}$[1]	151
AD[29]	32	AD[14]	72	VDD5	112	$\overline{\text{BE}}$[0]	152
VSSP2	33	AD[13]	73	DQ[14]	113	$\overline{\text{STROBE}}$	153
VDDP2	34	AD[12]	74	DQ[15]	114	ADR[14]	154
AD[28]	35	AD[11]	75	DQ[16]	115	ADR[13]	155
AD[27]	36	AD[10]	76	DQ[17]	116	ADR[12]	156
AD[26]	37	AD[9]	77	DQ[18]	117	ADR[11]	157
AD[25]	38	AD[8]	78	DQ[19]	118	ADR[10]	158
AD[24]	39	NC4	79	DQ[20]	119	ADR[9]	159
VDDP3	40	VSSP7	80	VSS7	120	VSS12	160

Memory Map

CY7C09449PVA resources are accessed from the PCI bus as an offset from Base Address Register 0 (BAR 0), unless otherwise indicated. Resources are also accessed from the local bus when the $\overline{\text{SELECT}}$ pin is active. PCI IO access to this memory map is also available through PCI IO pointers located at Base Address Register 1 (BAR 1). The memory map covers a continuous 32 KB address space.

PCI Bus

The CY7C09449PVA PCI bus operates according to the *PCI Specification Revision 2.2*. This section describes the specific PCI functions supported by the CY7C09449PVA. Reference URL: http://www.pcisig.com/

Table 8. Memory Map

Memory Contents	Address [14:0], Byte Offset	Size
I_2O Specific Registers	0x0000–0x03FF	1 KB
Operations Registers	0x0400–0x07FF	1 KB
Reserved	0x0800–0x1FFF	6 KB
Direct Access to PCI Bus (this is a window into PCI space; this window is only available to the local bus)	0x2000–0x3FFF	8 KB
Shared Memory	0x4000–0x7FFF	16 KB

Table 9. PCI Configuration Space

PCI Configuration Space				
31	**16**	**15**	**0**	**Address, Byte Offset**
Device ID, *RO*		Vendor ID, *RO*		0x00
Status, *CS*		Command, *CS*		0x04
Class Code, *RO*			Revision ID, *RO*	0x08
BIST (not used) 0x00	Header Type 0x00	Latency Timer, *RW*	Cache Line Size, *RW*	0x0C
Base Address Register #0—32KBytes Memory Space, *RW*				0x10
Base Address Register #1—8 Bytes IO Space, *RW*				0x14
Base Address Register #2 (not used) 0x0000				0x18
Base Address Register #3 (not used) 0x0000				0x1C
Base Address Register #4 (not used) 0x0000				0x20
Base Address Register #5 (not used) 0x0000				0x24
Cardbus CIS Pointer, *RO*				0x28
Subsystem Device ID, *RO*		Subsystem Vendor ID, *RO*		0x2C
Expansion ROM Base Address (not used) 0x0000				0x30
Reserved 0x0000				0x34
Reserved 0x0000				0x38
MAX_LAT, RO	MIN_GNT, RO	Interrupt Pin, RO	Interrupt Line, RW	0x3C

Legend for PCI Configuration Space	
0x00 or 0x0000	Hardwired to zero
RO	Read only: may be initialized by EEPROM across I^2C compatible serial interface
CS	Control and status register
RW	Read and write

Vendor ID

Address: 0x01 - 0x00

Default Value: 0x12BE

Read only: Initialized from the external memory accessed through the I^{2}C compatible serial interface

This two byte register contains the vendor ID assigned by the PCI SIG. The default value is the Cypress Semiconductor Vendor ID. Using the I^{2}C compatible serial interface for initialization provides a method to allow a manufacturer to load their own vendor ID.

Device ID

Address: 0x03 - 0x02

Default Value: 0x3042

Read only: Initialized from the external memory accessed through the I^{2}C compatible serial interface

This two byte register contains the device ID assigned by the manufacturer. The default value is the CY7C09449PVA chip device ID. Using the I^{2}C compatible serial interface for initialization provides a method to allow a manufacturer to load their own device ID.

Command

Address: 0x05 - 0x04

Default Value: 0x0000

Read and Write

This two byte register contains bits for device control. These bits are normally set by the system BIOS. The following bits are supported:

Bit 0:Enable response to IO space accesses.

Bit 1: Enable response to Memory space accesses.

Bit 2: Enable PCI bus master operation (may be initialized over the I^{2}C compatible serial interface).

Bit 3: Enable special cycle monitoring (CY7C09449PVA performs no special function as a target).

Bit 4: Enable bus master use the Memory Write and Invalidate command.

Bit 6: Enable the $\overline{\text{PERR}}$ signal for host notification of data parity errors.

Bit 8: Enable the $\overline{\text{SERR}}$ signal for host notification of system errors.

Bit 9: Enable fast back to back transactions to different agents (but CY7C09449PVA does not generate).

Status

Address: 0x07 - 0x06

Default Value: 0x0280

Read only and Write 1 to clear, except as indicated.

This two byte register contains bits for device status. The following bits are supported:

Bit 7: Read only bit set to indicate as a target, the chip accepts fast back to back transactions.

Bit 8: Set when $\overline{\text{PERR}}$ is asserted.

Bits 10 and 9: Read only bits set to 0x1 indicating medium response timing for DEVSEL.

Bit 12: Set when, as a master, the chip's transaction is terminated with Target-Abort.

Bit 13: Set when, as a master, the chip terminates a transaction with Master-Abort.

Bit 14: Set when $\overline{\text{SERR}}$ is asserted.

Bit 15: Set whenever a parity error is detected.

Revision ID

Address: 0x08

Default Value: 0x02

Read only: Initialized from the external memory accessed through the I^{2}C compatible serial interface.

This one byte register contains the Revision ID assigned by the manufacturer. The default value is set by Cypress Semiconductor at manufacturing time. Using the I^{2}C compatible serial interface for initialization provides a method to allow a manufacturer to load their own Revision ID.

Class Code

Address: 0x0B - 0x09

Default Value: 0x0E0001

Read only: Initialized from the external memory accessed through the I^{2}C compatible serial interface.

This three byte register contains the class code assigned by the manufacturer. The default value indicates an I$_2$O base class (0x0E), a sub class of 0x00, and the programming interface that supports system interrupt capability (0x01). Using the I^{2}C compatible serial interface for initialization provides a method to allow a manufacturer to load their own class code.

Cache Line Size

Address: 0x0C

Default Value: 0x00

Read and Write

This register contains the cache line size in DWORDs. The only valid size is 0x08; any other value written results in a 0x00 being written to the register. The value in this register is used to control when the master performs memory write and invalidate cycles. Additionally, the type of memory read command is determined by this value—Memory Read, Memory Read Line, or Memory Read Multiple.

Latency Timer

Address: 0x0D

Default Value: 0x00

Read and Write

This register controls how quickly the master must get off the bus if GNT is removed. The CY7C09449PVA implements bits [7:3] of this register, providing a granularity of eight clocks.

Base Address Register 0 (Memory Type Access)

Address: 0x13 - 0x10

Default Value: 0x00000000

Read all 32 bits, Write bits [31–15]

This register provides the base address of the CY7C09449PVA memory map. Bits [31–15] are read/write, indicating to the system BIOS that the shared memory space is 32 Kbytes. If a PCI memory transaction has address bits [31–15] matching the contents of this register and memory accesses are enabled (by command register bit 1), then the CY7C09449PVA chip acknowledges and accepts the transfer.

Base Address Register 1 (IO Type Access)

Address: 0x17 - 0x14

Default Value: 0x00000001

Read all 32 bits, Write bits [31–3]

This register provides the base address of the CY7C09449PVA IO pointer space. Bits [31–3] are read/write, indicating to the system BIOS that the IO pointer space is 8 bytes. If a PCI IO transaction has address bits [31–3] matching the contents of this and IO accesses are enabled (by command register bit 0), then the CY7C09449PVA acknowledges and accepts the transfer.

Cardbus CIS Pointer

Address: 0x2B - 0x28

Default Value: 0x00000000

Read only: Initialized from the external memory accessed through the I^{2}C compatible serial interface.

This register contains the Cardbus Card Information Structure (CIS). Using the I^{2}C compatible serial interface for initialization provides a method to allow a manufacturer to load their own CIS pointer value.

Subsystem Vendor ID

Address: 0x2D - 0x2C

Default Value: 0x0000

Read only: Initialized from the external memory accessed through the I^{2}C compatible serial interface.

This two byte register contains the subsystem vendor ID chosen by the manufacturer. Using the I^{2}C compatible serial interface for initialization provides a method to allow a manufacturer to load their own subsystem vendor ID.

Subsystem Device ID

Address: 0x2F - 0x2E

Default Value: 0x0000

Read only: Initialized from the external memory accessed through the I^{2}C compatible serial interface.

This two byte register contains the subsystem device ID chosen by the manufacturer. Using the I^{2}C compatible serial interface for initialization provides a method to allow a manufacturer to load their own subsystem device ID.

Interrupt Line

Address: 0x3C

Default Value: 0x00

Read and Write

This single byte register contains the interrupt line routing.

Interrupt Pin

Address: 0x3D

Default Value: 0x00

Read only: Initialized from the external memory accessed through the I^{2}C compatible serial interface.

This single byte register contains the interrupt pin information. The default value indicates that the CY7C09449PVA chip is not connected to the interrupts on the PCI bus. Using the I^{2}C compatible serial interface for initialization provides a method to allow a manufacturer to specify which interrupt pin is on the bus. Only bits [2–0] are implemented. All four interrupt numbers are supported, (INTA through INTD).

MIN_GNT

Address: 0x3E

Default Value: 0x00

Read only: Initialized from the external memory accessed through the I^{2}C compatible serial interface.

This single byte register contains the minimum grant time in 1/4 microsecond increments needed for efficient operation. The default value indicates that the add in card has no major requirements for setting the latency timer. The latency timer governs how long a burst transaction may use the PCI bus. Whatever the value, the CY7C09449PVA itself does not use the MIN_GNT data. It is used as a means to communicate system requirements to the host. Using the I^{2}C compatible serial interface for initialization provides a method to allow a manufacturer to load their own minimum grant time reflective of their add in card requirements.

MAX_LAT

Address: 0x3F

Default Value: 0x00

Read only: Initialized from the external memory accessed through the I^{2}C compatible serial interface.

This single byte register contains the minimum latency time in 1/4 microsecond increments needed for efficient operation. The default value indicates the add in card has no major requirements for how soon it needs access to the PCI bus after it has requested an access. Whatever the value, the CY7C09449PVA itself does not use MAX_LAT data. It is used as a means to communicate system requirements to the host. Using the I^{2}C compatible serial interface for initialization provides a method to allow a manufacturer to load their own minimum latency time reflective of their add in card requirements.

PCI Bus Commands

All memory and IO commands are supported as target and master.

■ IO Read	C/$\overline{\text{BE}}$[3:0] = 0x2
■ IO Write	C/$\overline{\text{BE}}$[3:0] = 0x3
■ Memory Read	C/$\overline{\text{BE}}$[3:0] = 0x6
■ Memory Write	C/$\overline{\text{BE}}$[3:0] = 0x7
■ Memory Read Multiple	C/$\overline{\text{BE}}$[3:0] = 0xC
■ Memory Read Line	C/$\overline{\text{BE}}$[3:0] = 0xE
■ Memory Write and Invalidate	C/$\overline{\text{BE}}$[3:0] = 0xF

All configuration commands are supported as target and master. Additionally, the CY7C09449PVA performs these accesses on its own PCI Configuration space. Control originates from the local bus using the CY7C09449PVA direct access feature. This is a necessary feature for the CY7C09449PVA to perform as a Host Bridge device. Type 0 and Type 1 PCI configuration commands are generated by the CY7C09449PVA. For details, see the Direct Access and Host Bridge descriptions in the section CY7C09449PVA Operations on page 48.

■ Configuration Read	C/$\overline{\text{BE}}$[3:0] = 0xA
■ Configuration Write	C/$\overline{\text{BE}}$[3:0] = 0xB

Interrupt Acknowledge and Special Cycle are supported on master cycles. As a target, no action is performed by the CY7C09449PVA.

■ Interrupt Acknowledge	C/$\overline{\text{BE}}$[3:0] = 0x0
■ Special Cycle	C/$\overline{\text{BE}}$[3:0] = 0x1

If the following command is not supported, a target access results in no response by the CY7C09449PVA as per the PCI specification.

■ Dual Address Cycle	C/$\overline{\text{BE}}$[3:0] = 0xD

The following commands are PCI Reserved and are not responded to according to PCI specification.

■ Reserved	C/$\overline{\text{BE}}$[3:0] = 0x4
■ Reserved	C/$\overline{\text{BE}}$[3:0] = 0x5
■ Reserved	C/$\overline{\text{BE}}$[3:0] = 0x8
■ Reserved	C/$\overline{\text{BE}}$[3:0] = 0x9

PCI IO Pointers

Utilization of PCI IO access is not generally recommended by the PCI special interest group. New system designs must use the PCI Memory access rather than PCI IO access. In general, this is provided as a support to legacy systems. The CY7C09449PVA Base Address Register 1 (BAR1) is the offset reference for PCI IO access to this device.

IO Address Pointer

Address: 0x1 - 0x0

Default Value: unknown, not initialized

Write only

The value written to this location is the offset into the CY7C09449PVA Memory Map. Bit 15 is “don’t care.”

IO Data Pointer

Address: 0x7 - 0x4

Default Value: unknown, not initialized

Read and Write

Upon a write to the pointer, the data is written to the location in the CY7C09449PVA Memory Map specified by the contents of the IO address pointer. If an IO read access to the pointer, then the data at the location in the CY7C09449PVA Memory Map, which is specified by the contents of the IO address pointer, is returned.

Local Bus

General Description

The CY7C09449PVA provides a configurable local processor bus interface which allows direct connection to several processor types. The interface is synchronous to the CLKIN signal. The CLKIN signal is tied to the local processor's clock, a derivative, or an independent clock source. To run the local interface at PCI clock speeds, any one of the PCLKOUT[2:0] pins must be connected to CLKIN.

The basic local processor bus transaction consists of an address phase, followed by one or more data phases. The interface signals are generally divided into those signals that qualify the address phase (ALE, STROBE, SELECT, READ, WRITE, and ADR[14:2]), and those that qualify data phases, (RDY_IN, RDY_IN, BLAST, BE[3:0], and DQ[31:0]). The CY7C09449PVA drives RDY_OUT to signal the need for wait states on the local processor bus and an indication of valid data on DQ[31:0] during read access of the CY7C09449PVA. Note that several of the CY7C09449PVA local bus signals have configurable polarity. These are: ALE, BLAST, RDY_OUT, and STROBE. Also, the READ and WRITE signals have special combined signal modes.

The basic local bus cycle starts with the address phase. The address phase is defined as both STROBE and SELECT active at the rising edge of CLKIN. Also sampled at this time are the READ and WRITE signals to determine if the access is a read or write. If the access is a read, then the CY7C09449PVA begins driving the DQ bus at the next CLKIN rising edge.

There are two ways to get an address into the CY7C09449PVA. With ALE tied active, the address is latched during the address phase. That is, when STROBE and SELECT are active, the address on the ADR[14:2] pins is latched on the rising edge of CLKIN. The second way is to use the trailing edge of ALE to latch the address. The CY7C09449PVA still needs a valid address phase (STROBE and SELECT active at the rising edge of CLKIN) before it begins processing the address. A valid and stable address must occur before the trailing edge of ALE and before the rising edge of CLKIN where STROBE and SELECT are active.

After the address phase come wait states and data phases. The STROBE signal is active or inactive during wait and data phases. A data phase occurs when the RDY_IN and RDY_IN inputs and the RDY_OUT output are all active at the rising edge of CLKIN. If any ready signal is inactive, then the next clock cycle is a wait state. The BE[3:0] pins are sampled during the data phase of write cycles to determine which data bytes are written. The data on the DQ pins are also latched at this time.

The BLAST signal is sampled during the data phase to determine if the last data phase occurs. In one mode, an inactive level during the data phase indicates that there are more data phases in the transaction and the address captured in the address phase must be updated. When BLAST is active during the data phase, it indicates that this is the last data phase of the transaction. In the other mode, BLAST is active during every data phase and goes inactive at the end of the last data phase. In both cases, if the access is a read, then the CY7C09449PVA stops driving the DQ bus synchronously with the rising edge of CLKIN for that data phase.

Interface Definitions

8-Bit Interface

The 8-bit interface option is selected by setting bits BW[1:0] = '00' in the Local Bus Configuration Register. Only data lines DQ[7:0] are used. The unused portion of the data bus, DQ[31:8] must be tied HIGH or LOW; the bits cannot be left floating. The least two significant bits of the local address bus must be connected to the byte enable pins BE[3:2]. BE[0] must be tied to RDY_IN which is connected normally.

Connect

- BE[3] = A1
- BE[2] = A0
- BE[1] = Logic HIGH
- BE[0] = Tie to RDY_IN

Table 10. 8-Bit Interface Option

BW[1:0]	BEMODE	A1, A0 (BE[3], BE[2])	DQ [7:0] Accessed Data
00	X	00	Data[7:0]
00	X	01	Data[15:8]
00	X	10	Data[23:16]
00	X	11	Data[31:24]

16-Bit Interface

The 16-bit interface option is selected by setting bits BW[1:0] = '01' in the Local Bus Configuration Register. Only data lines DQ[15:0] are used. The unused portion of the data bus, DQ[31:16] must be tied HIGH or LOW; the bits cannot be left floating. There are two basic modes for 16-bit operation. One is for Motorola style encoded byte enables and the other is for direct byte enables. This is configured with the Byte Enable mode bit, BEMODE. There is an exception to the data bus wiring for Motorola style buses if a 32-bit processor bus is configured to only use 16 data bits. The upper 16 bits of the processor bus are connected rather than the lower 16 bits. The description for BEMODE='1' follows.

BEMODE = '0' is for operation of other than Motorola style byte enables. Table 11 shows where data on the 16-bit bus is routed within the CY7C09449PVA internal data structures.

Connect

- BE[3] = A1
- BE[2] = Not used, must be tied HIGH
- BE[1] = BE1, UDS, BHE (Byte Enable 1, Upper Data Strobe, Byte High Enable)
- BE[0] = BE0, LDS, DEN (Byte Enable 0, Lower Data Strobe, Data Enable, A0)

Table 11. 16-Bit Interface Option

BW[1:0]	BE-MODE	A1 (BE[3])	DQ [15:0] Bus Accessed Data
01	0	0	CY7C09449PVA Data[15:0]
01	0	1	CY7C09449PVA Data[31:16]

Notes

BE[1:0] are used as byte enables. If the processor always does 16-bit accesses, then these are tied active LOW. These byte enables are also used for Upper Data Strobe (UDS) and Lower Data Strobe (LDS) for processors which produce these signals.

The least significant bit of the local address bus is tied to BE[3], and it *must be valid during the address phase.* This input must be incremented (toggled) at the end of each data phase. Bursts to the 16-bit interface do not need to start on a DWORD boundary. The internal DWORD address automatically increments after a data phase where BE[3] is HIGH.

BEMODE = '1' is for operation of Motorola style byte enables. Table 12 and Table 13 show where data on the 16-bit bus is placed in the CY7C09449PVA internal data structures. When a 32-bit Motorola processor bus is configured for 16-bit bus operation, connect the processor D[31:16] to CY7C09449PVA DQ[15:0]. For instance, the Motorola 68360 processor may be operated in this mode. The tables show this mode of operation.

Connect for encoded byte enables

- $\overline{\text{BE}}$[3] = SIZ1 (Operand Transfer Size, bit 1)
- $\overline{\text{BE}}$[2] = SIZ0 (Operand Transfer Size, bit 0)
- $\overline{\text{BE}}$[1] = A1
- $\overline{\text{BE}}$[0] = A0

Table 12. 16-Bit Interface Option (Motorola Style)

BW[1:0]	BEMODE	Interpretation
01	1	Use the following table

Table 13. 16-Bit Interface Option (Motorola Style)

CY7C09449PVA External $\overline{\text{BE}}$[3:0]	Interpretation ('byte' terminology here uses byte 3 as least significant byte of the processor's internal 32-bit data structure; the signals show pins on the processor.)	CY7C09449PVA Internal $\overline{\text{BE}}$[3:0] for writes
0000	all-byte write starting at byte 0 (D[31:16]) (truncated to two bytes)	0011
0001	all-byte write starting at byte 1 (D[23:16]) (truncated to one byte)	1011
0010	all-byte write starting at byte 2 (D[31:16]) (truncated to two bytes)	1100
0011	all-byte write starting at byte 3 (D[23:16]) (truncated to one byte)	1110
0100	single-byte write starting at byte 0 (D[31:24])	0111
0101	single-byte write starting at byte 1 (D[23:16])	1011
0110	single-byte write starting at byte 2 (D[31:24])	1101
0111	single-byte write starting at byte 3 (D[23:16])	1110
1000	two-byte write starting at byte 0 (D[31:16])	0011
1001	two-byte write starting at byte 1 (D[23:16]) (truncated to one byte)	1011
1010	two-byte write starting at byte 2 (D[31:16])	1100
1011	two-byte write starting at byte 3 (D[23:16]) (truncated to one byte)	1110
1100	three-byte write starting at byte 0 (D[31:16]) (truncated to two bytes)	0011
1101	three-byte write starting at byte 1 (D[23:16]) (truncated to one byte)	1011
1110	three-byte write starting at byte 2 (D[31:16]) (truncated to two bytes)	1100
1111	three-byte write starting at byte 3 (D[31:24]) (truncated to one byte) three-byte read starting at byte 3 (D[23:16]) (truncated to one byte)	1110

32-Bit Interface

The 32-bit interface option is selected by setting bits BW[1:0] = '10' or BW[1:0] = '11' in the Local Bus Configuration Register. Data lines DQ[31:0] are used. With BW[1:0] = '10', the byte enables are used directly as byte write enables. With BW[1:0] = '11', however, the meaning of the byte enables is determined from the following tables (based on BEMODE).

For 32-bit processor bus interfaces like the Motorola 68020 or 68030, BW = '11' and BEMODE = '0' settings are used. This supports a special style of using byte addressing instead of fully decoded byte enables. The SIZ1 and SIZ0 signals of the 68020 are connected to the $\overline{\text{BE}}$[3] and $\overline{\text{BE}}$[2] pins, respectively, and the A1 and A0 signals are connected to the $\overline{\text{BE}}$[1] and $\overline{\text{BE}}$[0] pins on the CY7C09449PVA.

Connect for encoded byte enables

- $\overline{\text{BE}}$[3] = SIZ1 (Operand Transfer Size, bit 1)
- $\overline{\text{BE}}$[2] = SIZ0 (Operand Transfer Size, bit 0)
- $\overline{\text{BE}}$[1] = A1
- $\overline{\text{BE}}$[0] = A0

Table 14. 32-Bit Interface option

BW[1:0]	BEMODE	Interpretation
10	X	Use byte enables for all 4 byte lanes
11	0	Use the following table

Table 15. 32-Bit Interface Option (Motorola Style)

CY7C09449PVA External $\overline{\text{BE}}$[3:0]	Interpretation 'byte' terminology here uses byte 3 as least significant byte of the processor's internal 32-bit data structure; the signals show pins on the processor	CY7C09449PVA Internal $\overline{\text{BE}}$[3:0] for writes
0000	all-byte write starting at byte 0 (D[31:0])	0000
0001	all-byte write starting at byte 1 (D[23:0]) (truncated to three bytes)	1000
0010	all-byte write starting at byte 2 (D[15:0]) (truncated to two bytes)	1100
0011	all-byte write starting at byte 3 (D[7:0]) (truncated to one byte)	1110
0100	single-byte write starting at byte 0 (D[31:24])	0111
0101	single-byte write starting at byte 1 (D[23:16])	1011
0110	single-byte write starting at byte 2 (D[15:8])	1101
0111	single-byte write starting at byte 3 (D[7:0])	1110
1000	two-byte write starting at byte 0 (D[31:16])	0011
1001	two-byte write starting at byte 1 (D[23:8])	1001
1010	two-byte write starting at byte 2 (D[15:0])	1100
1011	two-byte write starting at byte 3 (D[7:0]) (truncated to one byte)	1110
1100	three-byte write starting at byte 0 (D[31:8])	0001
1101	three-byte write starting at byte 1 (D[23:0])	1000
1110	three-byte write starting at byte 2 (D[15:0]) (truncated to two bytes)	1100
1111	three-byte write starting at byte 3 (D[7:0]) (truncated to one byte)	1110

For 32-bit processor bus interfaces like the Motorola 68040, BW = '11' and BEMODE = '1' settings are used. This supports a special style of using byte addressing instead of fully decoded byte enables. The SIZ1 and SIZ0 signals of the 68040 are connected to the BE[3] and BE[2] pins respectively. The A1 and A0 signals are connected to the BE[1] and BE[0] pins on the CY7C09449PVA. A cache line fill is triggered using the SIZ1 and SIZ0 pins on the 68040 type bus. When these bits are set to '11', the CY7C09449PVA interprets this as a burst of four, ignoring the burst last signal BLAST.

Table 16. 32-Bit Interface Option (Motorola Style)

BW[1:0]	BEMODE	Interpretation
11	1	Use the following table

Table 17. 32-Bit Interface Option (Motorola Style)

CY7C09449PVA External	Interpretation	CY7C09449PVA Internal
$\overline{\text{BE}}$[3:0]	**The signals show pins on the processor**	**$\overline{\text{BE}}$[3:0] for writes**
00xx	32-bit write D[31:0]	0000
0100	8-bit write D[31:24]	0111
0101	8-bit write D[23:16]	1011
0110	8-bit write D[15:8]	1101
0111	8-bit write D[7:0]	1110
100x	16-bit write D[31:16]	0011
101x	16-bit write D[15:0]	1100
11xx[1]	burst of four 32-bit writes, $\overline{\text{BLAST}}$ not used	0000

Note

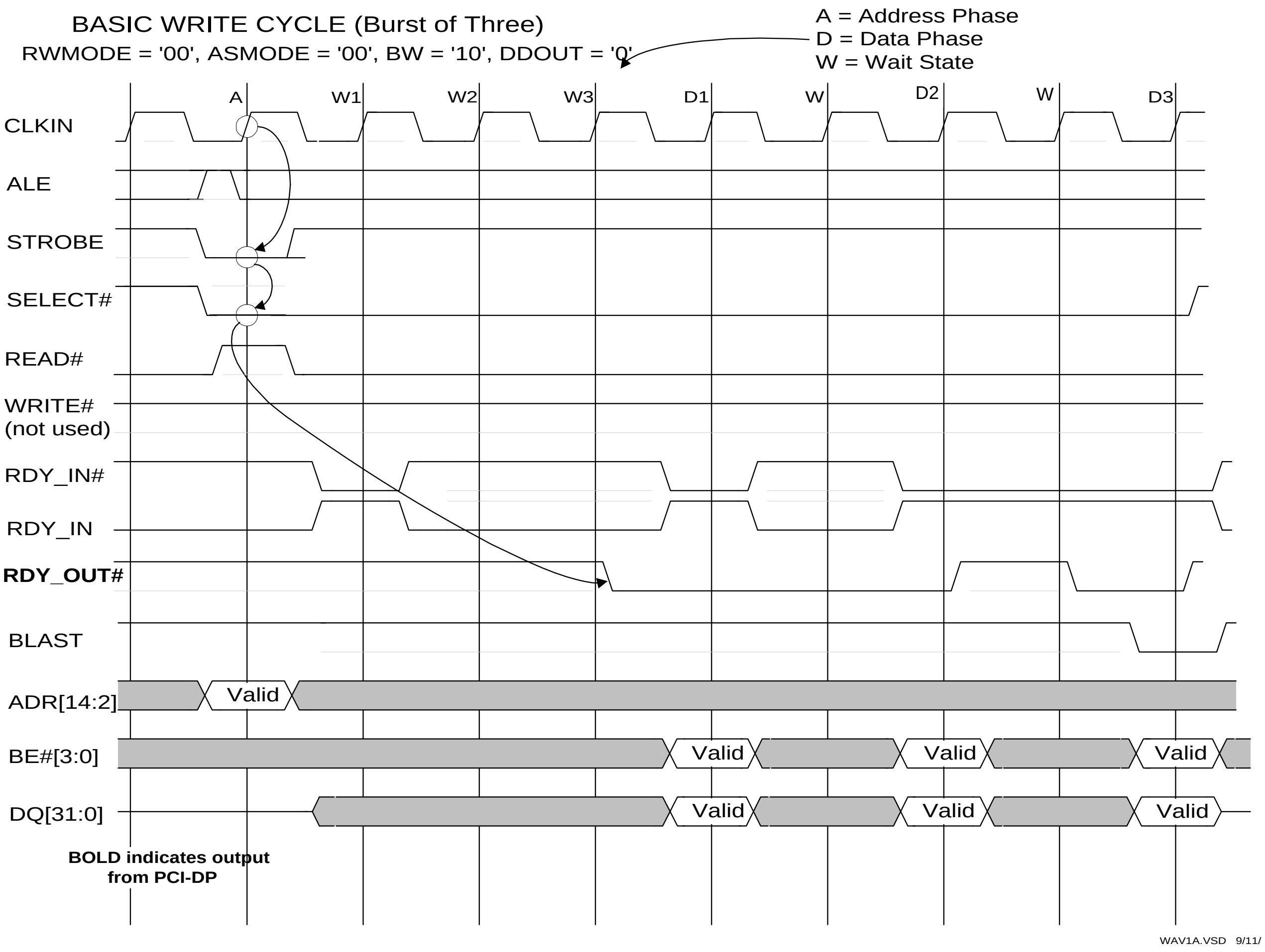
[1] This encoding, {BW[1:0], BEMODE, BE[3:2]} = {'11111'}, results in a burst of four DWORD. $\overline{\text{BLAST}}$ must remain active.

Timing Diagrams

Write Cycle

A basic write cycle is illustrated in Figure 3. It includes a burst of three data phases on a 32-bit wide bus.

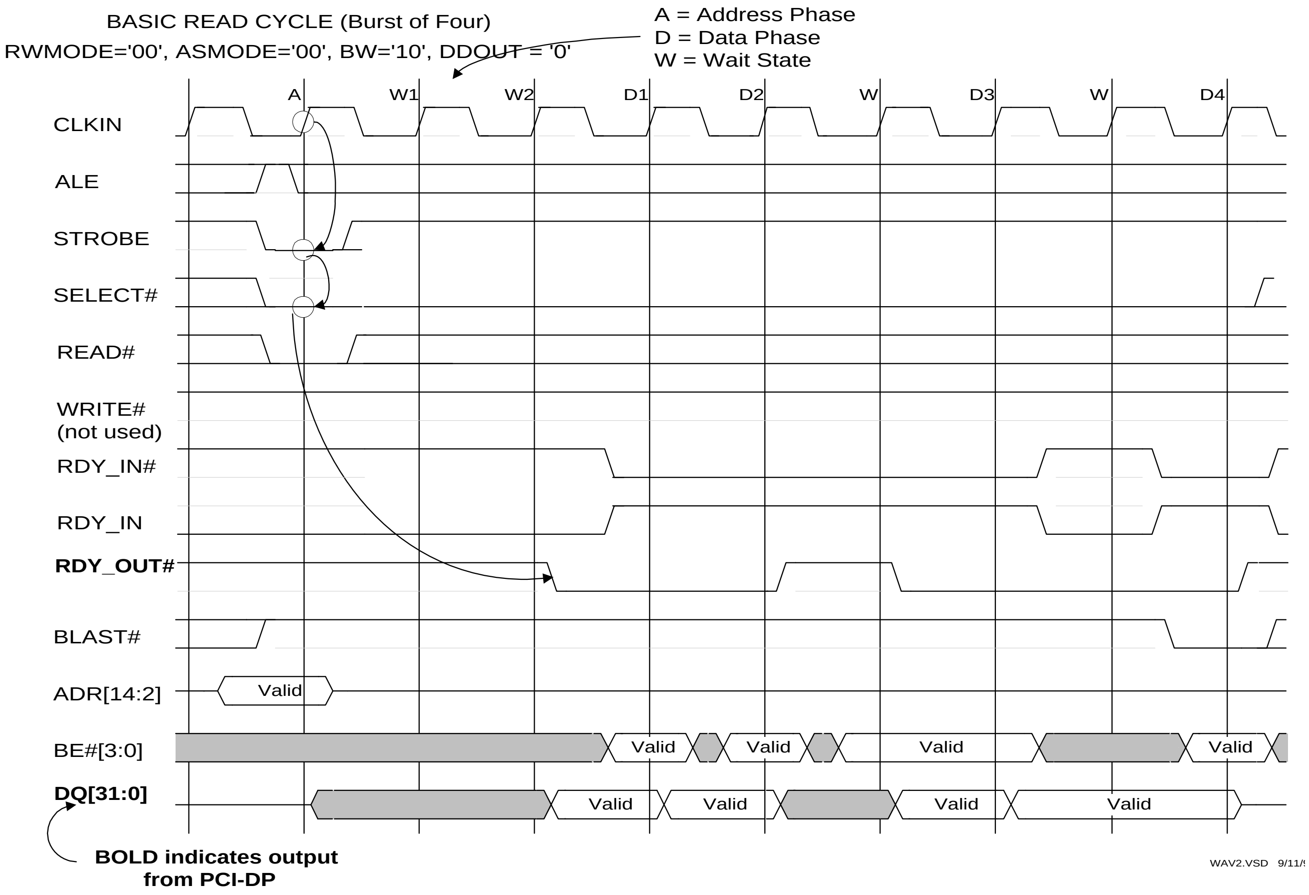
Figure 3. Basic Write Cycle (Burst of Three)



Read Cycle

The basic read cycle differs from the write cycle only in the level of the $\overline{\text{READ}}$ and $\overline{\text{WRITE}}$ signals, and the timing and driving of the data bus DQ. A basic read burst of four data phases on a 32-bit wide bus is illustrated in Figure 4.

Figure 4. Basic Read Cycle (Burst of Four)



Basic 8-Bit Interface

Figure 5 and Figure 6 illustrate the operation of the 8-bit interface mode. Note that only data lines DQ[7:0] are used. DQ[31:8] are not used and must be tied high or low; they cannot be left floating. The least significant bits of the local address, A[1] and A[0], must be connected to the byte enable pins BE[3] and BE[2], respectively. These must be valid during the address phase.

In burst operation, BE#[3:2] are inputs used at A1 and A0 of the local address bus. Bursts to the 8-bit interface do not need to start on a DWORD boundary. The internal DWORD address automatically increments after a data phase where BE[3:2] equals '11', (A[1:0] = '11').

Figure 5. Single Cycle Operation

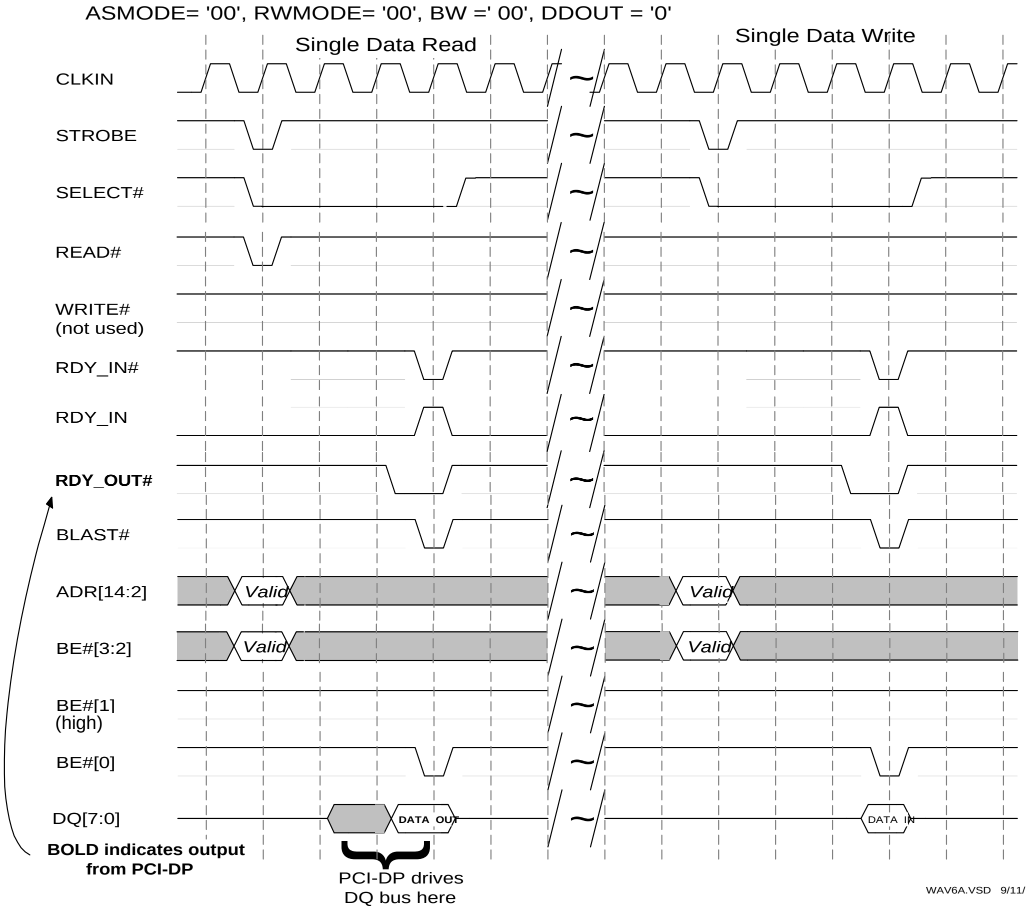
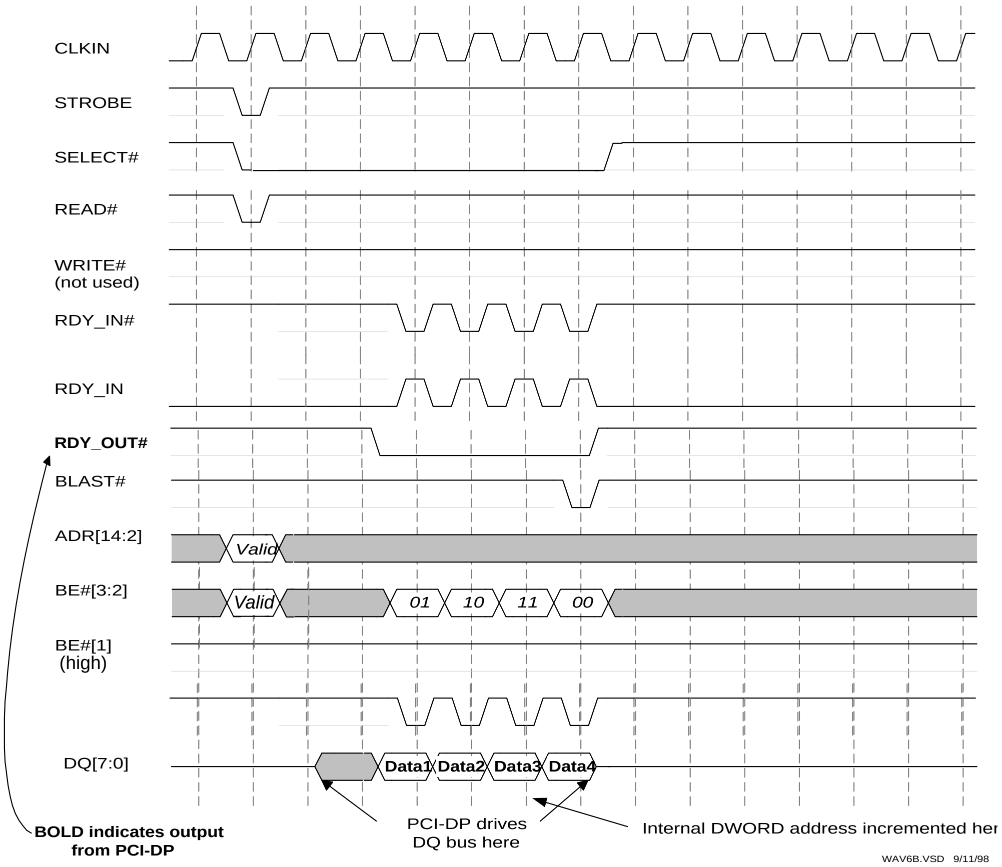


Figure 6. Data Burst Operation

ASMODE = '00', RWMODE = '00', BW = '00', DDOUT = '0'



BOLD indicates output from PCI-DP

PCI-DP drives DQ bus here

Internal DWORD address incremented her

WAV6B.VSD   9/11/98

Basic 16-Bit Interface

The following two waveforms illustrate the operation of the 16-bit interface mode. Note that only data lines DQ[15:0] are used. DQ[31:16] are unused and must be tied HIGH or LOW; they cannot be left floating. The least significant bit of the local address of the 16-bit bus, A[1], must be connected to the byte enable pin $\overline{BE}$[3]. It must be valid during the address phase.

Note that $\overline{BE}$[1:0] are used as byte enables. If the processor always does 16-bit accesses, then these are tied active LOW. These byte enables are also used for Upper Data Strobe (UDS) and Lower Data Strobe (LDS) for processors which produce these signals.

In burst operation, $\overline{BE}$[3] must be incremented (toggled) at the end of each data phase. Bursts to the 16-bit interface do not need to start on a DWORD boundary. The internal DWORD address automatically increments after a data phase where $\overline{BE}$[3] equals '1', (A[1] = '1').

Figure 7. Single Cycle Operation

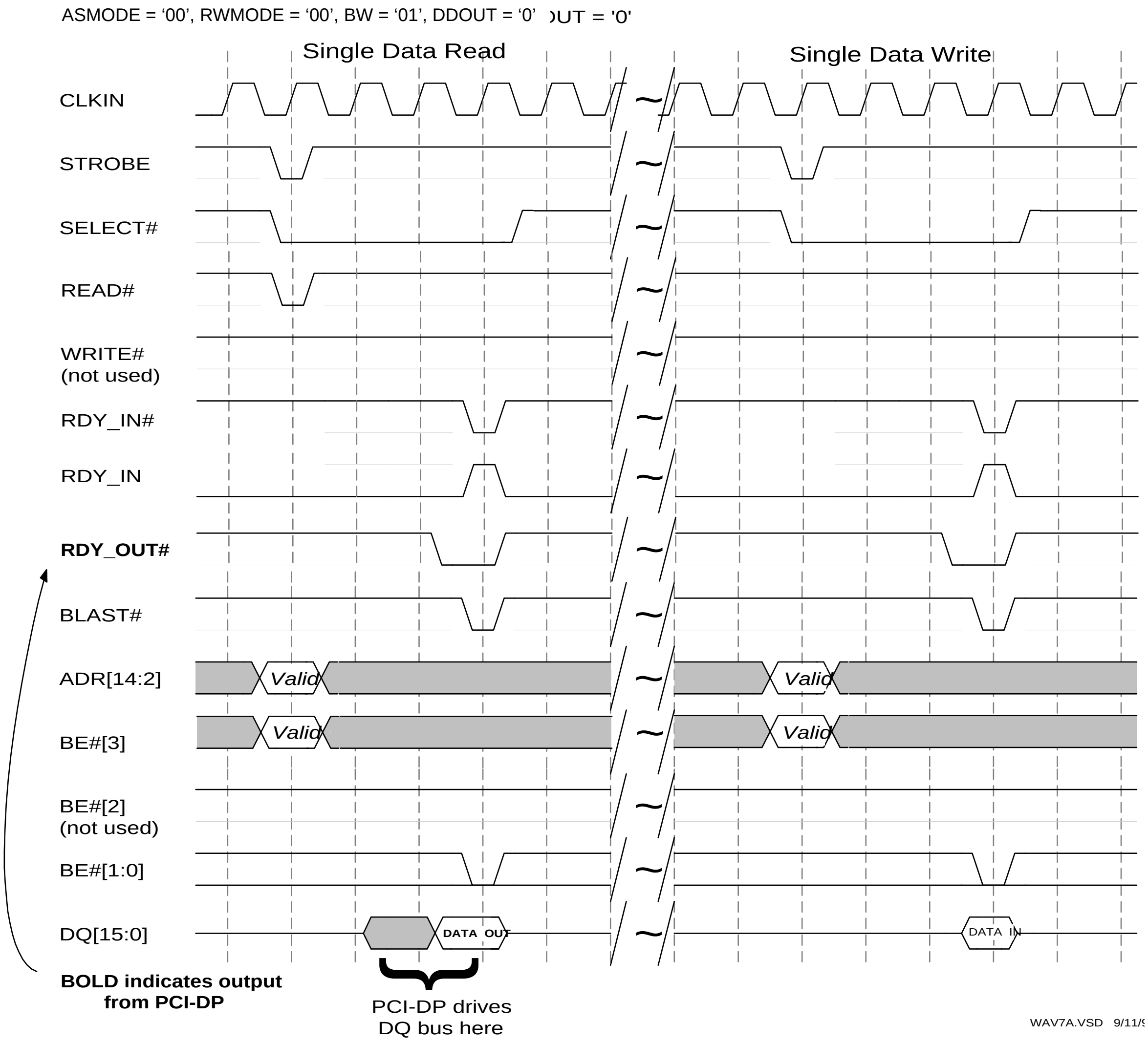
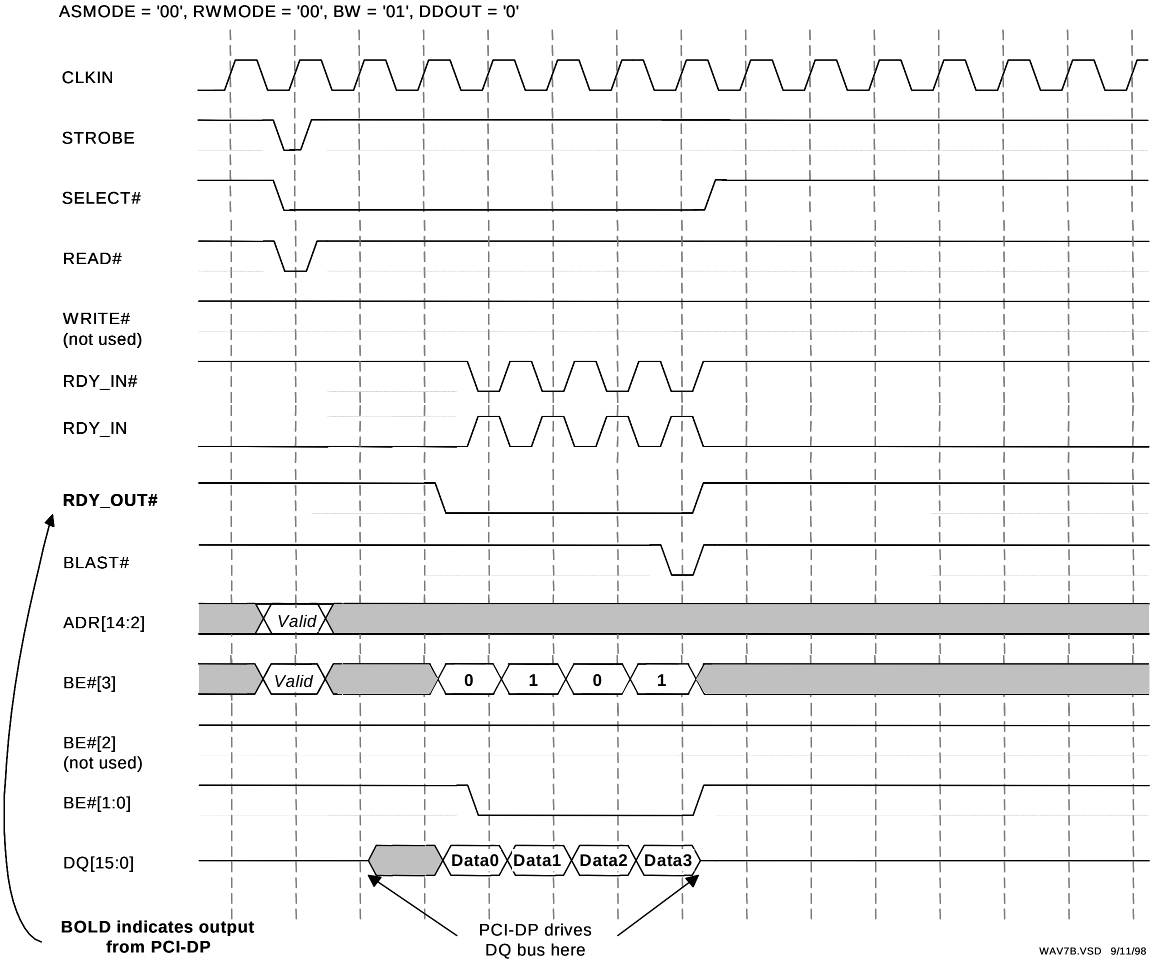


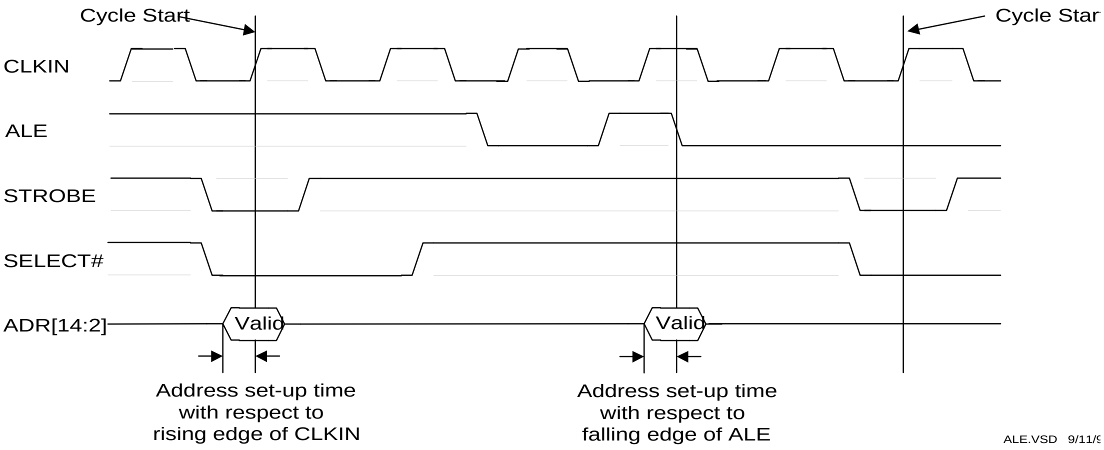
Figure 8. Data Burst Operation



ALE—Address Latch Enable

The ALE signal may be used in two modes. With ALE tied active, the address is latched during the address phase. That is, when the STROBE and SELECT signals are active, the address on the ADR[14:2] pins is latched on the rising edge of CLKIN. The second way is to use the trailing edge of ALE to latch the address. The CY7C09449PVA still needs a valid address phase (STROBE and SELECT active at the rising edge of CLKIN) before it begins processing the address. A valid and stable address must occur before the trailing edge of ALE and before the rising edge of CLKIN where STROBE and SELECT are active. The active polarity of ALE is defined in the Operations Registers: ALE_POL of the Local Bus Configuration Register.

Figure 9. ALE Operation Modes


Cycle Start
Cycle Star
CLKIN
ALE
STROBE
SELECT#
ADR[14:2]
Valid
Valid
Address set-up time with respect to rising edge of CLKIN
Address set-up time with respect to falling edge of ALE
ALE.VSD 9/11/9

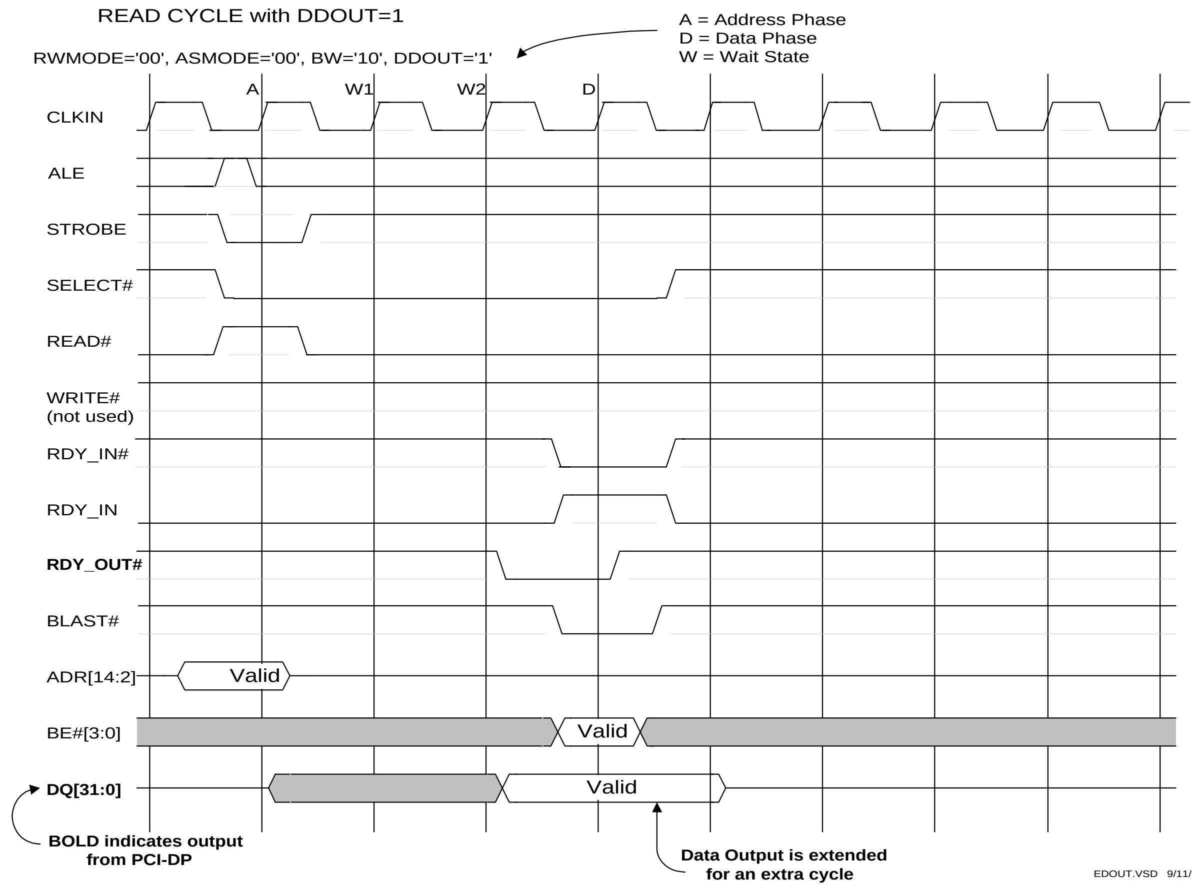
RDY_OUT_OE—Ready Out Tri-state Mode

The RDY_OUT signal may be configured to drive at all times or to tri-state when inactive. The tri-state mode is a sustained deasserted function. In tri-state mode, when RDY_OUT is to go inactive, RDY_OUT is driven to the deasserted level for one clock and then tri-stated. It remains tri-stated until RDY_OUT is asserted. The logic polarity of RDY_OUT is programmable. The mode (RDY_OUT_OE) and polarity (RDYOUT_POL) controls are set in the Local Bus Configuration Register, LBUSCFG, of the Operations Registers.

DDOUT—Delayed Data Out

The delayed data out control defines when the CY7C09449PVA drives the DQ bus during a local bus read. The control is defined in the Operations Registers: DDOUT of the Local Bus Configuration Register. When DDOUT = '0', the CY7C09449PVA drives the DQ bus during a read starting one clock after the address phase and stop driving at the clock edge where both of the ready inputs and BLAST# are active. When DDOUT = '1', the CY7C09449PVA drives the DQ bus during a read starting one CLKIN clock after the address phase and stop driving one clock after the clock edge where the two ready inputs and BLAST are active. The data is driven for one clock period after the signal that the transaction is over. In the case of multiple data phases, it adds one clock cycle to the starting latency of the burst.

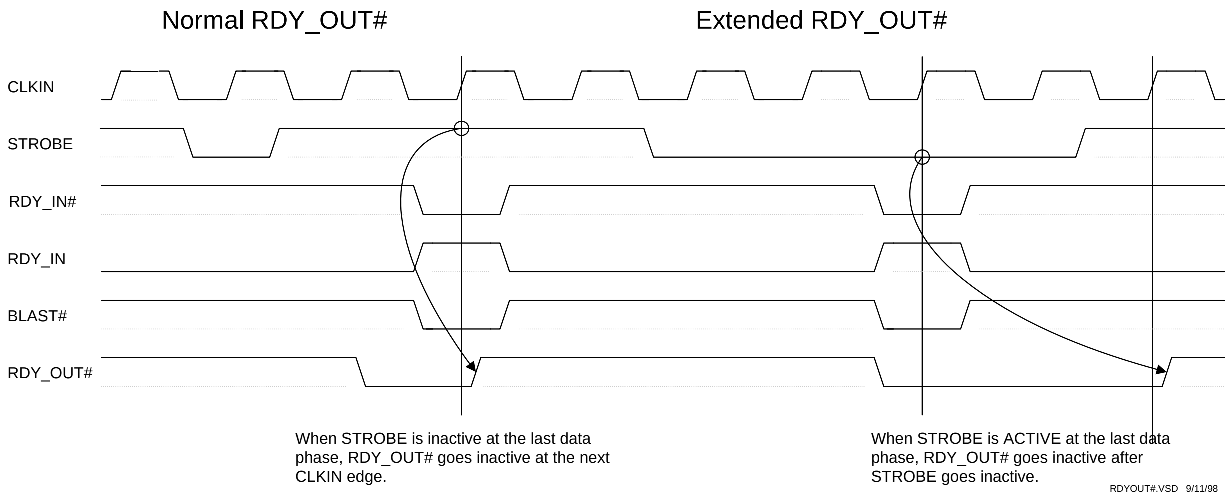
Figure 10. Read Cycle with Delayed Data Out



RDY_OUT—Ready Out

STROBE is active or inactive during data phases. However, if STROBE is active during the data phase when BLAST is active and the extended ready out control (XTND_RDY_OUT) is set, the CY7C09449PVA keeps RDY_OUT active until STROBE goes inactive. In the case of a read, the CY7C09449PVA continues to drive the data on DQ until STROBE is deasserted.

Figure 11. Ready Out Operation (STROBE Controlled)



When STROBE is inactive at the last data phase, RDY_OUT# goes inactive at the next CLKIN edge.

When STROBE is ACTIVE at the last data phase, RDY_OUT# goes inactive after STROBE goes inactive.

LINE_WRAP_DIS—Cache Line Wrap Disable

This setting is used to disable cache line wrapping, LINE_WRAP_DIS = ‘1’. Cache line wrapping occurs only when the local bus interface is set for 32 bit width with encoded byte enables, using the Motorola byte enable encoding, and the bus SIZ bits indicating a cache line access. Specifically, this is when BW = '11', BEMODE = '0', and BE[3] = BE[2] = ‘1.’ If a cache line access is made and cache line wrapping is disabled, then the burst proceeds linearly with no implicit address wraparound at the four DWORD boundary.

ASMODE—Address Strobe Mode

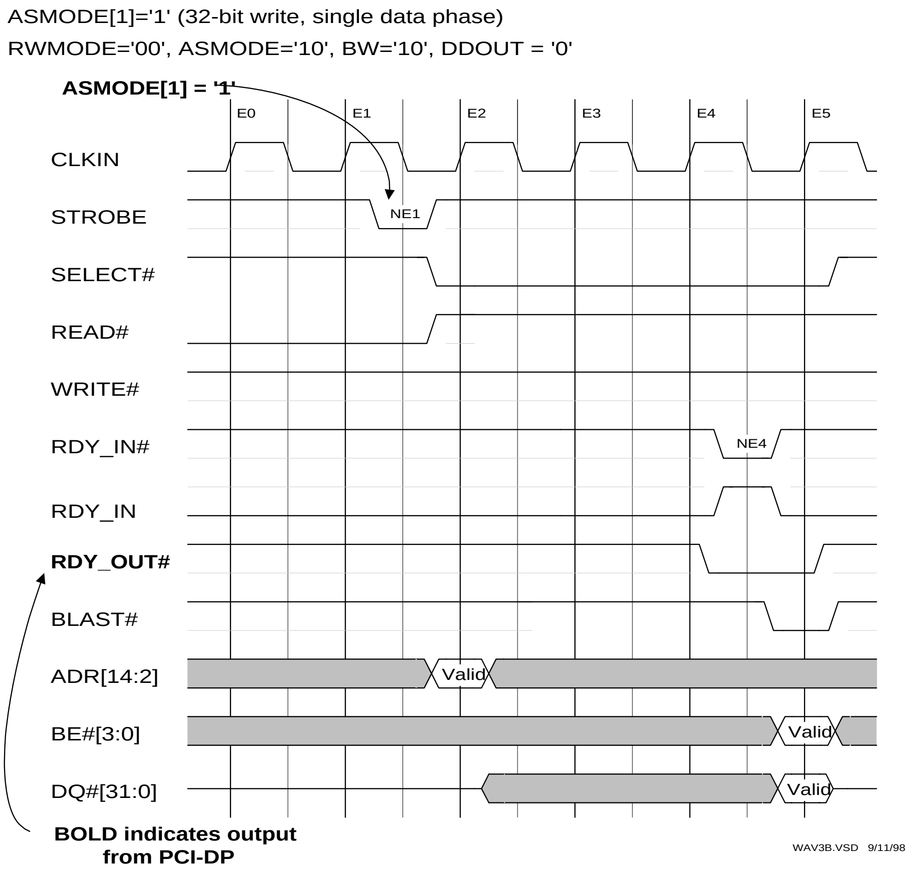
The address strobe mode control defines the polarity and the timing used to sample the CY7C09449PVA address strobe input signal, STROBE. The two bit control field is defined in the Operations Registers: ASMODE of the Local Bus Configuration Register.

ASMODE[0] defines the polarity of the STROBE input signal; '0' = active LOW and '1' = active HIGH.

ASMODE[1] controls the sampling edge of the STROBE signal. Logic LOW indicates that the signal is sampled using the rising edge of CLKIN. A logic HIGH indicates that the signal is sampled with the falling edge of CLKIN. Sampling on the falling edge must only be used when the required minimum setup time with respect to the clock rising edge cannot be met on the signals.

The following waveform illustrates the operation of the ASMODE[1] pin. ADR and READ are sampled at E2, and a valid write occurs at E4. STROBE is captured at the negative edge labeled NE1 and the ADR and READ signals are sampled at the positive clock edge labeled E2. The ready signals are also sampled at the negative edge labeled NE4 and not at the positive edge labeled E5. STROBE is active LOW since ASMODE[0] = '0'.

Figure 12. 32-Bit Write, Single Data Phase



ASMODE[1]='1' (32-bit write, single data phase)

RWMODE='00', ASMODE='10', BW='10', DDOUT = '0'

BOLD indicates output from PCI-DP

WAV3B.VSD 9/11/98

Falling Edge Sampling for RDY_IN, RDY_IN, SELECT, and STROBE

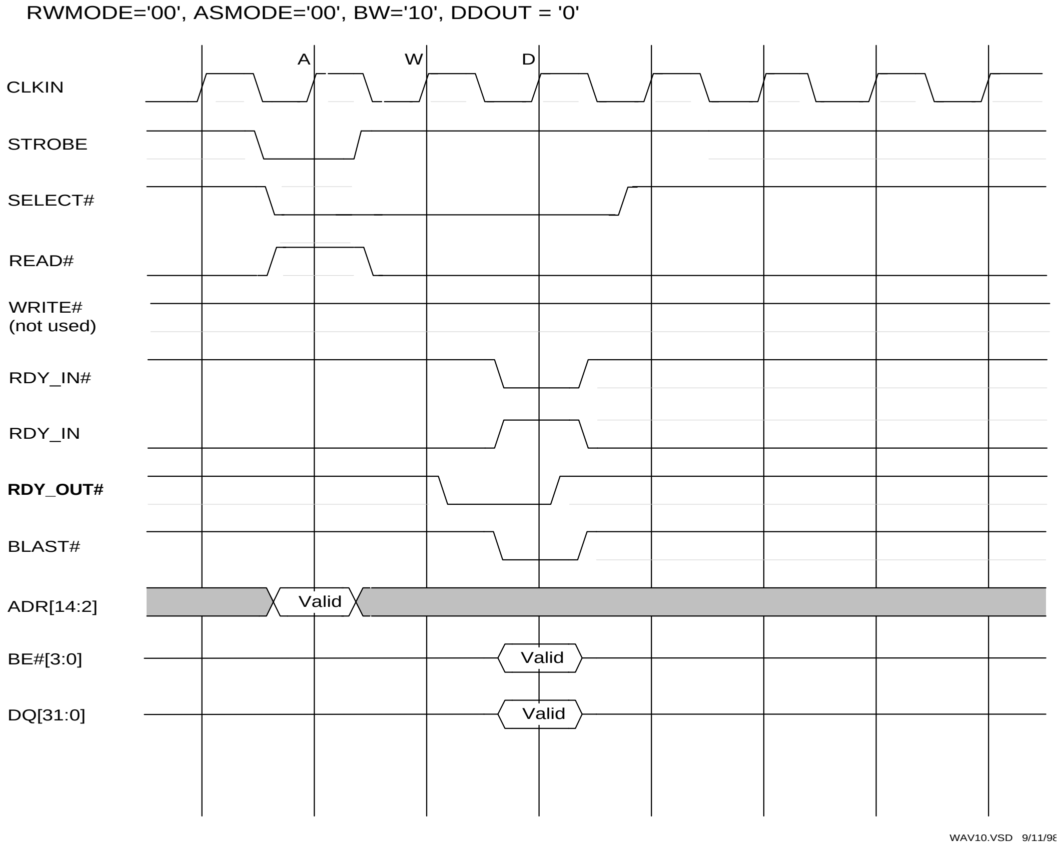
These signals may be configured for falling edge sampling within the Local Bus Configuration Register (LBUSCFG) of the Operations Registers. RDY_IN and RDY_IN sampling is configured by the RDY_IN_FALL bit, SELECT sampling is configured by the SELECT_FALL bit, and STROBE sampling is configured by the ASMODE[1] bit. Setting any one or all of these bits does not affect the sampling of other signals on the local bus. That is, all other signals that are synchronous inputs are sampled on the rising edge of the local bus clock, CLKIN. When a negative edge sample is used, the other signals are qualified by that sample on the immediately following rising edge of CLKIN. For example, study the prior waveform illustrating operation of ASMODE. In that diagram, STROBE is configured to sample on the falling edge of CLKIN because ASMODE[1]='1'. An active STROBE indicates an address phase. The valid address is captured on the first rising clock edge after STROBE is sampled active.

RWMODE—Read Write Mode

The read write mode control defines how the address strobe ($\overline{\text{STROBE}}$), read ($\overline{\text{READ}}$), and write ($\overline{\text{WRITE}}$) input signals are interpreted by the CY7C09449PVA internal logic. The two bit control field is defined in the Operations Registers: RWMODE of the Local Bus Configuration Register. Each of the four cases for RWMODE are illustrated in Figure 13 through Figure 15.

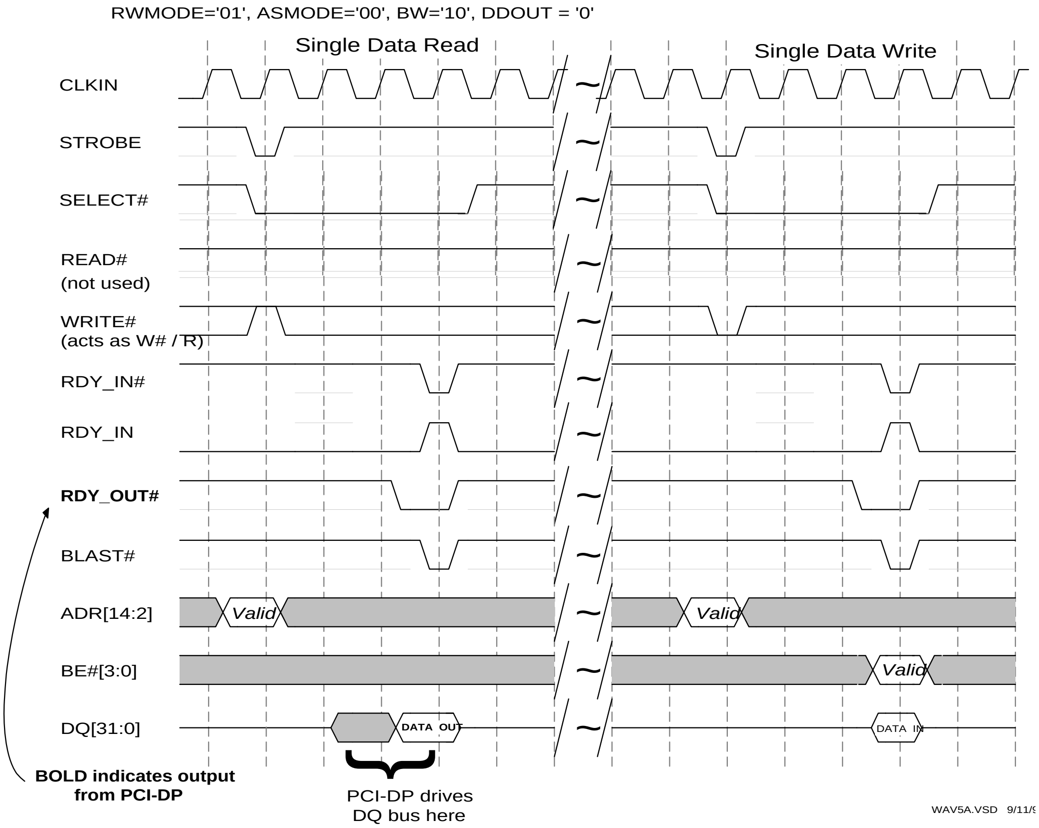
Use RWMODE = '00' to interface to a processor that has a read write signal defined as W_$\overline{\text{R}}$ (write is logic 1, read is logic 0). In this mode, the $\overline{\text{WRITE}}$ is not used and must be tied HIGH. Figure 13 illustrates a write cycle.

Figure 13. Read Write Mode (RWMODE = '00')



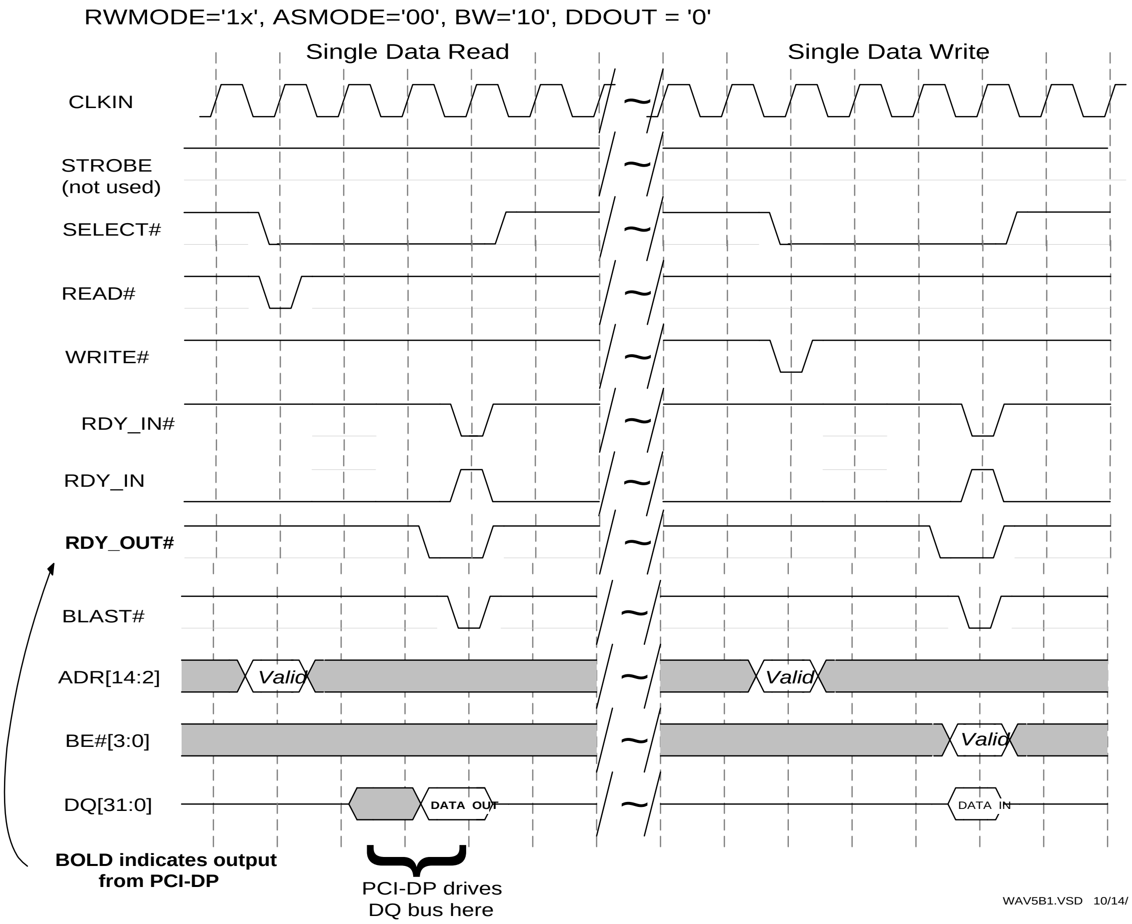
Use RWMODE = '01' to interface to a processor that has a read write signal defined at R_W (write is logic 0, read is logic 1). In this mode, the WRITE acts as R_W and it is sampled when SELECT and STROBE are both active. The READ pin is not used and must be tied HIGH. This is illustrated in Figure 14.

Figure 14. Read Write Mode (RWMODE = '01')



Use RWMODE = '10' or RWMODE = '11' to interface to a processor that has separate active LOW read and write signals. The two modes are identical. Sampling of the READ and WRITE signals is used as the internal address strobe in place of the STROBE signal. This is illustrated in Figure 15.

Figure 15. Read Write Mode (RWMODE = '1x')



I^{2}C Serial Port and Auto Configuration

The CY7C09449PVA I^{2}C serial port may master the I^{2}C bus, but it is not a target on the bus. Read and write access to the port is available to both the PCI and local buses through the I^{2}C programming Operations Registers. The CY7C09449PVA supports single byte device internal addressing. The port is used for auto configuration of the CY7C09449PVA and for basic read and write access to I^{2}C compatible devices connected to the port.

Auto configuration is the function that uses the port to load CY7C09449PVA configuration information. A typical device containing the data is a serial Electrically Erasable Programmable Read Only Memory (EEPROM). The EEPROM includes data for some PCI configuration registers and some Operations Registers. The EEPROM containing the CY7C09449PVA configuration data must be located at I^{2}C device address 0x0 and must contain the proper CY7C09449PVA Signature. For details, see Table 18 and the accompanying field descriptions.

Table 18. I^{2}C Serial Port Device 0x0 Memory Map for Auto Configuration[2]

Byte 3 don't care	**Byte 2** don't care	**Byte 1** don't care	**Byte 0** don't care	**Internal Address, Byte Offset** 0x00 ... 0x3F
CY7C09449PVA Signature 0x48	**CY7C09449PVA Signature** 0x37	reserved	reserved	0x40
Device ID high byte	**Device ID** low byte	**Vendor ID** high byte	**Vendor ID** low byte	0x44
Class Code, base class high byte	**Class Code,** sub class middle byte	**Class Code,** programming intf. low byte	**Revision ID**	0x48
Subsystem Device ID high byte	**Subsystem Device ID** low byte	**Subsystem Vendor ID** high byte	**Subsystem Vendor ID** low byte	0x4C
MAX_LAT	**MIN_GNT**	**Interrupt Pin, Master Enable**[3]	don't care	0x50
Cardbus CIS Pointer high byte	**Cardbus CIS Pointer** low byte	**Cardbus CIS Pointer** high byte	**Cardbus CIS Pointer** low byte	0x54
reserved	reserved	reserved	reserved	0x58
reserved	reserved	reserved	reserved	0x5C
reserved	reserved	reserved	reserved	0x60
reserved	reserved	reserved	reserved	0x64
reserved	reserved	reserved	reserved	0x68
reserved	**Local Bus Configuration** high byte	**Local Bus Configuration** middle byte	**Local Bus Configuration** low byte	0x6C
reserved	reserved	reserved	reserved	0x70
Host Control bits [31:24]	**Host Control** bits [23:16]	**Host Control** bits [15:8]	**Host Control** bits [7:0]	0x74
don't care	don't care	don't care	don't care	0xFF … 0x78

Notes

[2] The recommended value for reserved data in the EEPROM is '1'.

[3] Master Enable is the most significant bit of this byte; see text for more description of this flag.

CY7C09449PVA Signature

Address: 0x43 - 0x42

Device Configuration Signature: A valid EEPROM CY7C09449PVA configuration image is indicated at this address by the value of 0x4837. It is read from the EEPROM at I^{2}C device address 0x0 immediately after the CY7C09449PVA comes out of reset. The CY7C09449PVA comes out of reset as indicated by the deassertion of the CY7C09449PVA RST input. Upon recognition of a valid signature, the contents of the EEPROM is transferred to the appropriate CY7C09449PVA registers. The appropriate registers are indicated by the other labeled fields of the I^{2}C Serial Port Device 0x0 Memory Map for auto configuration and are described in this section. If the value at this location is not 0x4837, then the transfer does not occur and the default (reset) values for the CY7C09449PVA registers remain in effect after the CY7C09449PVA comes out of reset.

Vendor ID

Address: 0x45 - 0x44

PCI Configuration Vendor ID: the meaning of this field is described in the section PCI Bus on page 8.

Device ID

Address: 0x47 - 0x46

PCI Configuration Device ID: the meaning of this field is described in the section PCI Bus on page 8.

Revision ID

Address: 0x48

PCI Configuration Revision ID: the meaning of this field is described in the section PCI Bus on page 8.

Class Code

Address: 0x4B - 0x49

PCI Configuration Class Code (Base Class, Sub Class, Programming Interface): the meaning of this field is described in the section PCI Bus on page 8.

Subsystem Vendor ID

Address: 0x4D - 0x4C

PCI Configuration Subsystem Vendor ID: the meaning of this field is described in the section PCI Bus on page 8.

Subsystem Device ID

Address: 0x4F - 0x4E

PCI Configuration Subsystem Device ID: the meaning of this field is described in the section PCI Bus on page 8.

Interrupt Pin

Address: 0x51, bits 2, 1, 0

PCI Configuration Interrupt Pin: the meaning of this field is described in the section PCI Bus on page 8.

Master Enable

Address: 0x51, bit 7

PCI Configuration Command Bit 2: enables PCI bus master operation. For a host bridge, this typically must be set to allow the host to configure itself and configure and access other devices on the PCI bus. Even though an external master can manipulate the PCI Command register, it is typical that the host is the first device to configure devices on the PCI bus. Since the default value for PCI Command bit 2 is that PCI bus mastering is disabled, the Master Enable bit in the EEPROM image must be set to enable PCI mastering.

MIN_GNT

Address: 0x52

PCI Configuration MIN_GNT: the meaning of this field is described in the section PCI Bus on page 8.

MAX_LAT

Address: 0x53

PCI Configuration MAX_LAT: the meaning of this field is described in the section PCI Bus on page 8.

Cardbus CIS Pointer

Address: 0x57 - 0x54

PCI Configuration Cardbus CIS Pointer: the meaning of this field is described in the section PCI Bus on page 8.

Local Bus Configuration

Address: 0x6C - 0x6E

Operations Registers Local Bus Configuration: the detailed meaning of this field is described in the section Operations Registers on page 31. For the CY7C09449PVA local bus to exhibit the correct protocol, the Local Bus Configuration Operations Register is loaded before the CY7C09449PVA is accessed through the local bus interface. The local bus interface circuitry is held in reset until transfer of the EEPROM configuration data is complete. At such completion, and dependent upon the state of the Host Control Operations Register, the local bus is available for access using the programmed local bus interface protocol.

Host Control

Address: 0x77 - 0x74

Operations Registers Host Control: Only bits 1 and 0 have meaning; the other bits are reserved. When programming bits 1 and 0, other bits of the DWORD must be written with '0'.

One of two reset controls from the CY7C09449PVA may be used to reset the local processor system. The CY7C09449PVA RSTOUT output signal is a buffered copy of the PCI bus RST signal and is not conditioned by the bits of the Host Control register. This signal deasserts before the auto configuration process completes, so some applications do not use this signal to prevent premature local processor attempts to access the CY7C09449PVA.

The other form of reset control provides a direct link to the auto configuration process. Using this method, the local processor remains in reset until completion of the auto configuration process. In this case, the CY7C09449PVA RSTOUTD output signal (or its active HIGH version, RSTOUTD) is used to reset the local processor. This signal is a copy of bit 0 of the Host Control register (the Local Processor Reset). With auto configuration, the local processor is either held in reset or released from reset depending upon the value in the EEPROM. Furthermore, as an Operations Register, the Host Control register may be accessed from the PCI bus after auto configuration is complete. Therefore, if to hold the local processor in reset until updated by a command over the PCI bus, bit 0 of this field must be set.

Bit 1 of the Host Control register is cleared in most cases. Setting it to '1' resets the Operations Registers to their default state and thereby reinitialize the Local Bus Configuration register. This bit is typically used only for debug or maintenance operations. Another seldom used operation is setting the Host Control register from the local processor. Even though the Operations Register is available from the local bus, setting either bit 0 or bit 1 to '1' locked out the local processor from accessing the CY7C09449PVA by way of the local bus interface.

Operations Registers

These registers are the means by which CY7C09449PVA functions are accessed. Access is available through either interface, the PCI bus or the local bus. The Operations Registers include the PCI Bus Mastering registers (DMA), the I_2O messaging unit registers, the interrupt registers, the mail boxes, and the direct access register. Also included in the Operations Registers are the initialization and configuration registers used to customize the CY7C09449PVA operation to the user's needs. The critical Host Control Register and Local Bus Configuration Register may be programmed during the system initialization process. Programming through the I^2C serial is available for this purpose.

The Operations Registers reside in the local bus clock domain, therefore, a clock must be applied to CLKIN for proper operation of the CY7C09449PVA. Either an external clock may be provided or one of the CY7C09449PVA PCLKOUT[2:0] signal outputs may be used. PCLKOUT is a copy of the PCI Clock input, CLK. The PCLKOUT signals are intended as an option for the user to connect to other circuits as well.

Also available to support other circuits, the user may connect any or all of the three reset outputs from the CY7C09449PVA. RSTOUT is a registered copy of the PCI Reset input, RST. It is synchronous to CLKIN. The other two reset output signals, RSTOUTD and RSTOUTD, are the copy of a bit in an Operations Register, the 'R' bit of the Host Control Register. RSTOUTD and RSTOUTD complement each other and are synchronous to CLKIN. Upon power up reset of the CY7C09449PVA (through the PCI Reset RST) this bit is set active. It may be cleared during the startup process using the I^2C serial interface. It may also be cleared or set through commands received over the PCI bus. Therefore, these signals are used to hold a local processor in reset until CY7C09449PVA configuration is complete or when a host is ready to release the local processor to begin its operations.

Operations Registers Addresses

This is a summary table of the CY7C09449PVA Operations Registers. Register locations are the offset from the Base Address Register 0 and are DWORD aligned. The value shown is the address of the least significant byte of the register offset. Default, power up values are also shown. Both numbers are documented in hexadecimal notation. Bit positions in gray are unused and read back as '0' unless otherwise indicated in the default value.

Table 19. Operations Registers

<table>
<tr><th colspan="13">Operations Register</th><th>Offset / Mnemonic</th></tr>
<tr><th colspan="2">31 24</th><th>23 16</th><th colspan="5">15 8</th><th colspan="5">7 0</th><th>Default Value</th></tr>
<tr><td colspan="13"><b>I_2O Host Interrupt Status Register</b></td><td>0x0030 I2OHISR</td></tr>
<tr><td colspan="2"></td><td></td><td colspan="5"></td><td></td><td>I</td><td></td><td></td><td></td><td>0x00000000</td></tr>
<tr><td colspan="13"><b>I_2O Host Interrupt Mask Register</b></td><td>0x0034 I2OHIMR</td></tr>
<tr><td colspan="2"></td><td></td><td colspan="5"></td><td></td><td>M</td><td></td><td></td><td></td><td>0xFFFFFFFF</td></tr>
<tr><td colspan="13"><b>I_2O Local Interrupt Status Register</b></td><td>0x0038 I2OLISR</td></tr>
<tr><td colspan="2"></td><td></td><td colspan="5"></td><td></td><td>I</td><td></td><td></td><td></td><td>0x00000000</td></tr>
<tr><td colspan="13"><b>I_2O Local Interrupt Mask Register</b></td><td>0x003C I2OLIMR</td></tr>
<tr><td colspan="2"></td><td></td><td colspan="5"></td><td></td><td>M</td><td></td><td></td><td></td><td>0xFFFFFFFF</td></tr>
<tr><td colspan="8"><b>I_2O FIFO Access</b></td><td colspan="6">all default as empty FIFO, read as 0xFFFFFFFF</td></tr>
<tr><td colspan="13">Inbound Free FIFO (read only) and Inbound Post FIFO (write only)</td><td>0x0040 IBFPFIFO</td></tr>
<tr><td colspan="13">Outbound Post FIFO (read only) and Outbound Free FIFO (write only)</td><td>0x0044 OBPFFIFO</td></tr>
<tr><td colspan="13">Inbound Post FIFO (read only) and Inbound Free FIFO (write only)</td><td>0x0048 IBPFFIFO</td></tr>
<tr><td colspan="13">Outbound Free FIFO (read only) and Outbound Post FIFO (write only)</td><td>0x004C OBFPFIFO</td></tr>
<tr><td colspan="13"><b>Direct Access</b></td><td>0x0460 DAHBASE</td></tr>
<tr><td colspan="4">PCI Physical Base Address (4 Gbyte, 8 Kb blocks)</td><td></td><td>F</td><td></td><td>A1A0</td><td>BE for Reads</td><td></td><td colspan="2">PCI</td><td></td><td>0xXXXXXXXX</td></tr>
<tr><td colspan="13"><b>I^2C Serial Command Register (write only)</b></td><td>0x04A0 NVCMD</td></tr>
<tr><td></td><td>Device Address</td><td>Memory Address</td><td colspan="5">Write Data</td><td colspan="3"></td><td>T</td><td>R</td><td>n/A</td></tr>
<tr><td colspan="13"><b>I^2C Serial Read Data Register</b></td><td>0x04A4 NVREAD</td></tr>
</table>

Table 19. Operations Registers (continued)

Operations Register				**Offset / Mnemonic**
31 24	**23 16**	**15 8**	**7 0**	**Default Value**
Byte 3	Byte 2	Byte 1	Byte 0	0xXXXXXXXX
I^{2}C Serial Status Register				0x04A8 NVSTAT
			ACK D	0x000000XX
DMA Local Base Address Register				0x04B0 DMALBASE
		Local Base Address (16 Kbyte)		0x0000XXXX
DMA Host Base Address Register				0x04B4 DMAHBASE
PCI Base Address (4 Gbyte)				0xXXXXXXXX
DMA Burst Size Register				0x04B8 DMASIZE
		DMA Burst Size (16K byte)		0x0000XXXX
DMA Control Register				0x04BC DMACTL
		L P	PI W	0x0000000X
Arbitration Utility Flag Register				0x04C0 ARBUTIL
L3 P3	L2 P2	L1 P1	L0 P0	0x00000000
Host Control				0x04E0 HCTL
			S R	0x00000001
Host Interrupt Control/Status				0x04E4 HINT
Interrupt Enable		Interrupt Status		0x00000000
Host to Local Data Mailbox				0x04E8 HLDATA
I		Byte 1	Byte 0	0x0000XXXX
Local Processor Interrupt Control/Status				0x04F4 LINT
Interrupt Enable		Interrupt Status		0x00000000
Local to Host Data Mailbox				0x04F8 LHDATA
I		Byte 1	Byte 0	0x0000XXXX
Local Bus Configuration				0x04FC LBUSCFG
	Local Bus Configuration			0x00010B50

Operations Registers Descriptions

Detailed descriptions of the Operations Registers follow. Register locations are the offset from the Base Address Register 0 and are DWORD aligned. The value shown is the address of the least significant byte of the register offset. The offsets are documented in hexadecimal notation. Unused bits are grayed out. Unused bits are read as '0' unless otherwise indicated.

I_2O Registers

I_2O Host Interrupt Status Register—I2OHISR0x0030

31																												3			0
																												I			

Bit	Description
3 -- I	Interrupt from the Outbound Post FIFO—the FIFO is not empty. This bit is continuously updated to reflect the status of the FIFO. It is read only; ‘0’: no interrupt; ‘1’: interrupt signalled. '0' default. **Note** Unused bits in this register are read as 0s.

I_2O Host Interrupt Mask Register—I2OHIMR0x0034

31																												3			0
																												M			

Bit	Description
3 -- M	Host Interrupt Mask Bit 1: interrupt is masked (default) 0: interrupt is not masked **Note** Unused bits in this register are read as 1s.

I_2O Local Interrupt Status Register—I2OLISR0x0038

31																												3			0
																												I			

Bit	Description
3 -- I	Interrupt from the Inbound Post FIFO—the FIFO is not empty. This bit is continuously updated to reflect the status of the FIFO. It is read only; ‘0’: no interrupt; ‘1’: interrupt signalled. '0' default. **Note** Unused bits in this register are read as 0s.

I_2O Local Interrupt Mask Register—I2OLIMR0x003C

31																												3			0
																												M			

Bit	Description
3 -- M	Local Interrupt Mask Bit 1: interrupt is masked (default) 0: interrupt is not masked **Note** Unused bits in this register are read as 1s.

I_2O Inbound Free and Post FIFO—IBFPFIFO0x0040

31	0
Inbound Free FIFO (read only) and Inbound Post FIFO (write only)	

Bit	Description
31:0	A shared port—Reading from this port returns data from the Inbound Free FIFO. The read of an empty FIFO returns 0xFFFF FFFF. Writing to this port places data into the Inbound Post FIFO. If the FIFO is already full, the contents of the FIFO do not change; the data written is lost. The FIFO is initially empty. An asserted RST# empties all CY7C09449PVA FIFO; all data is lost.

I_2O Outbound Post and Free FIFO—OBPFFIFO0x0044

31	0
Outbound Post FIFO (read only) and Outbound Free FIFO (write only)	

Bit	Description
31:0	A shared port—Reading from this port returns data from the Outbound Post FIFO. The read of an empty FIFO returns 0xFFFF FFFF. Writing to this port places data into the Outbound Free FIFO. If the FIFO is already full, the contents of the FIFO do not change; the data written is lost. The FIFO is initially empty. An asserted $\overline{\text{RST}}$ empties all CY7C09449PVA FIFO; all data is lost.

I_2O Inbound Post and Free FIFO—IBPFFIFO0x0048

31	0
Inbound Post FIFO (read only) and Inbound Free FIFO (write only)	

Bit	Description
31:0	A shared port—Reading from this port returns data from the Inbound Post FIFO. The read of an empty FIFO returns 0xFFFF FFFF. Writing to this port places data into the Inbound Free FIFO. If the FIFO is already full, the contents of the FIFO do not change; the data written is lost. The FIFO is initially empty. An asserted $\overline{\text{RST}}$ empties all CY7C09449PVA FIFO; all data is lost.

I_2O Outbound Free and Post FIFO—OBFPFIFO0x004C

31	0
Outbound Free FIFO (read only) and Outbound Post FIFO (write only)	

Bit	Description
31:0	A shared port—Reading from this port returns data from the Outbound Free FIFO. The read of an empty FIFO returns 0xFFFF FFFF. Writing to this port places data into the Outbound Post FIFO. If the FIFO is already full, the contents of the FIFO do not change; the data written is lost. The FIFO is initially empty. An asserted $\overline{\text{RST}}$ empties all CY7C09449PVA FIFO; all data is lost.

Direct Access Register

Direct Access Host Physical Base Address Register—DAHBASE0x0460

31 13		11		9 8	7 4		2 1	0
PCI Physical Base Address (4G byte, 8K byte blocks)		F		A1A0	Byte Enables for Reads		Type	

Bit	Description
31:13	PCI Physical Base Address specifying 8 Kbyte block
11 -- F	When '1', force contents of A1A0 to PCI during the PCI address phase.
9:8	Value placed on PCI bus, PCI A1 = bit 9, PCI A0 = bit 8.
7:4	Data Byte Enables for PCI Master Reads, C/BE#[3:0].
2:1 Type	PCI command cycle type for PCI Master Access 00 = interrupt acknowledge (read) (PCI command 0x0) or special cycle (write) (PCI command 0x1) 01 = IO cycle (read or write) (PCI command 0x2 or 0x3) 10 = memory cycle (read or write) (PCI command 0x6 or 0x7) 11 = configuration cycle (read or write) (PCI command 0xA or 0xB)

I^2C Serial Port Registers

I^2C Serial Command Register—NVCMD (a write only register)0x04A0

31	30 24	23 16	15 8	7	1	0
	Device Address	Memory Address	Write Data		T	R

Bit	Description
30:24	**Device Address.** Device address of the I^2C serial device. Default is 1010000.
23:16	**Memory Address.** Address within the I^2C serial device.
15:8	**Write Data.** Write data. This data is ignored if the command is a read.
1 -- T	**Read Type.** This bit is ignored if the command is a write. The data read from a I^2C serial device is accessible from the NVREAD register. 1 = 4-byte read 0 = single byte read
0 -- R	Read / Write 1 = read command 0 = write command **Note** The write of this byte triggers the start of the EEPROM access. In an 8 or 16-bit system, this location must be written after the address and data have been written.

I^{2}C Serial Read Data Register—NVREAD0x04A4

This register contains one or four bytes of data read from the I^{2}C serial EEPROM.

31 24	23 16	15 8	7 0
Byte 3	Byte 2	Byte 1	Byte 0

Bit	Description
31:24 Byte 3	Stores sequential read, byte 3. Undefined for single byte read.
23:16 Byte 2	Stores sequential read, byte 2. Undefined for single byte read.
15:8 Byte 1	Stores sequential read, byte 1. Undefined for single byte read.
7:0 Byte 0	Stores single read and sequential read, byte 0.

I^{2}C Serial Status Register—NVSTAT0x04A8

This register contains status information about the I^{2}C serial data transfer.

31 8	7 5	4	0
	ACK		D

Bit	Description
7:5 -- ACK	**Acknowledge** bit 7 = device address ack bit. 0 = ack, 1 = no ack. bit 6 = address ack bit. 0 = ack, 1 = no ack. bit 5 = second address ack bit. 0 = ack, 1 = no ack. In a successful read or write, these bits are 000.
0 -- D	**Done Indicator** 1 = done; 0 = in progress

PCI Bus Mastering (DMA) Registers

DMA Local Base Address Register—DMALBASE0x04B0

31 14	13 2	0
	Local Base Address (16K byte)	

Bit	Description
13:2	Local Base Address: This is the first address of the DMA in the local memory. This register has DWORD resolution.

DMA Host Physical Base Address Register—DMAHBASE0x04B4

31 2	0
PCI Physical Base Address (4 Gbyte)	

Bit	Description
31:2	PCI Physical Base Address—This is the first address of the DMA in the host's memory space. This register has DWORD resolution.

DMA Size Register—DMASIZE0x04B8

31 14	13 2	0
	DMA Burst Size (16K byte)	

Bit	Description
13:2	Burst Size for any mastered DMA, read or write. This register has DWORD resolution.

DMA Control Register—DMACTL0x04BC

31 10	9	8	7 2	1	0
	L	P		PI	W

Bit	Description
9 -- L	Local Ownership: Writing to this bit by the local processor updates the value if and only if the P bit is not set to ‘1.’ A write to this bit from the PCI bus never updates this bit. This bit (along with the P bit) is intended to facilitate software arbitration of the DMA registers. '0' default.
8 -- P	PCI Ownership: Writing to this bit by the PCI bus updates the value if and only if the L bit is not set to ‘1’. A write to this bit from the local processor never updates this bit. This bit (along with the L bit) is intended to facilitate software arbitration of the DMA registers. '0' default.
1 -- PI	Pre-Fetch Inhibit for PCI memory reads: When this bit is set to one, the CY7C09449PVA PCI bus master engine only uses the PCI Read command (0x6). When this bit is zero (default), the CY7C09449PVA PCI bus master engine uses PCI commands Read (0x6), Read Line (0xC), and Read Multiple (0xE) as appropriate to optimize utilization of the PCI system bus(es).
0 -- W	Write: Determines the direction of the DMA and starts transfer. 1: DMA operation is a write to the PCI bus memory from the CY7C09449PVA shared memory. 0: DMA operation is a read from the PCI bus memory into the CY7C09449PVA shared memory. A write to the low byte of this register triggers the DMA to occur. '0' default.

Arbitration Utility Flag Register—ARB_FLAGS0x04C0

31	25	24	23 18	17	16	15 10	9	8	7 2	1	0
	L3	P3		L2	P2		L1	P1		L0	P0

Bit	Description
25 -- L3	L3 Ownership: A write to this bit by the local processor updates this bit if and only if the P3 bit is not set to ‘1’. '0' default.
24 -- P3	P3 Ownership: A write to this bit by the PCI bus updates this bit if and only if the L3 bit is not set. '0' default.
17 -- L2	L2 Ownership: A write to this bit by the local processor updates this bit if and only if the P2 bit is not set to ‘1’. '0' default.
16 --P2	P2 Ownership: A write to this bit by the PCI bus updates this bit if and only if the L2 bit is not set. '0' default.
9 -- L1	L1 Ownership: A write to this bit by the local processor updates this bit if and only if the P1 bit is not set to ‘1’. '0' default.
8 -- P1	P1 Ownership: A write to this bit by the PCI bus updates this bit if and only if the L1 bit is not set. '0' default.
1 -- L0	L0 Ownership: A write to this bit by the local processor updates this bit if and only if the P0 bit is not set to ‘1’. '0' default.
0 -- P0	P0 Ownership: A write to this bit by the PCI bus updates this bit if and only if the L0 bit is not set. '0' default.

Host Control and Status Registers

Host Control Register—HCTL0x04E0

31	2	1	0
		S	R

Bit	Description
1 -- S	**Soft Reset** – This bit controls the internal reset for the CY7C09449PVA. 1 = reset active 0 = not reset (default)
0 -- R	**Local Processor Reset** – This bit controls the $\overline{\text{RSTOUTD}}$ and RSTOUTD pins. 1 = reset active (default state, RSTOUTD pin is LOW, RSTOUTD pin is HIGH) 0 = not reset (RSTOUTD pin is HIGH, RSTOUTD pin is LOW)

This register contains two types of reset bits. The Local Processor Reset bit, R, is intended for use by circuitry connected to the CY7C09449PVA local bus. The output signals RSTOUTD and $\overline{\text{RSTOUTD}}$ are a direct reflection of the state of this bit. The RSTOUTD (or its active low version $\overline{\text{RSTOUTD}}$) is connected to the local processor system's reset. R is set to '1' when the CY7C09449PVA reset input is asserted (RST = '0'). After deassertion of $\overline{\text{RST}}$, R remains set to '1' until the CY7C09449PVA auto configuration process is complete and it is cleared either through the auto configuration termination control setting or through the PCI bus interface. If RSTOUTD is not used as the local processor system reset, then it is also cleared through the local bus interface.

The second reset bit is the Soft Reset bit, S. This is used to reset certain internal registers and states of the CY7C09449PVA. It is primarily intended for test and debug operations during product development. When S = '1,' it resets all of the Operations Registers according to their reset default values with the following exceptions:

- DMACTL (at offset 0x04BC): All bits are reset to '0', except bits PL and W remain unchanged
- HINT (at offset 0x04E4): All bits remain unchanged
- LINT (at offset 0x04F4): Bit 3 (Host to Local Mailbox) is cleared to '0', all other bits remain unchanged
- LBUSCFG (at offset 0x04FC): All bits remain unchanged

Additional behavior when S = '1' is as follows:

- DQ[31:00] is held at high impedance;
- Local bus state machine is held in idle;
- DMA state machine is held in idle;
- PCI bus state machine mastering access is held in idle; and
- FIFO are emptied and flags return to default (empty).

Host Interrupt Control and Status Register—HINT0x04E4

31 25	24 16	15 10	9 0
	Interrupt Enable		Interrupt Status

Bit	Description
25:16 Interrupt Enable	**Interrupt Enables** 0000000000 =no interrupts are enabled (default) xxxxxxxxx1 =I_2O Local FIFO overflow interrupt enabled xxxxxxxx1x =I_2O PCI FIFO overflow interrupt enabled xxxxxxx1xx =reserved; always read as 0 xxxxxx1xxx =Local to host, mailbox interrupt enabled xxxxx1xxxx =Local to host, external signal interrupt enabled xxxx1xxxxx =DMA complete interrupt enabled xxx1xxxxxx =I_2O inbound post FIFO not empty interrupt enabled xx1xxxxxxx =I_2O outbound post FIFO not empty interrupt enabled x1xxxxxxxx =PCI target abort interrupt enabled 1xxxxxxxxx =PCI master abort interrupt enabled **Note** All enable bits are initially cleared.
9:0 Interrupt Status	**Interrupt Event Status** 0000000000 = no events active xxxxxxxxx1 =I_2O Local FIFO overflow xxxxxxxx1x =I_2O PCI FIFO overflow xxxxxxx1xx =reserved; always read as 0 xxxxxx1xxx =Local to host mailbox xxxxx1xxxx =Local to host external signal interrupt xxxx1xxxxx =DMA complete xxx1xxxxxx =I_2O inbound post FIFO not empty (mirror of I2OLISR[3]; read only at this address) xx1xxxxxxx = I_2O outbound post FIFO not empty (mirror of I2OHISR[3]; read only at this address) x1xxxxxxxx = PCI target abort 1xxxxxxxxx = PCI master abort **Note** When an event status bit is active, writing a '1' to that bit location clears the bit except for bits 6 and 7. All event status bits are initially cleared.

Host to Local Data Mailbox—HLDATA0x04E8

31 25	24	23 16	15 8	7 0
	I		Byte 1	Byte 0

Bit	Description
24 -- I	Interrupt to local This bit enables the host to send an interrupt to the local. When it is set to 1 by the host, it triggers a mailbox interrupt to the local processor. The interrupt remains active until it is cleared by writing to the Local Interrupt Control and Status Register (LINT). 0 = inactive 1 = active This bit is write only.
15:8 -- Byte 1 7:0 -- Byte 0	Data byte Two bytes of data that are written by the host and read by the local processor.

Local Control and Status Registers

Local Interrupt Control and Status Register—LINT0x04F4

31 25	24 16	15 10	9 0
	Interrupt Enable		Interrupt Status

Bit	Description
25:16 Interrupt Enable	Interrupt Enables 0000000000 = no interrupts are enabled (default) xxxxxx xxx1 = I_2O Local FIFO overflow interrupt enabled xx xxxx xx1x = I_2O PCI FIFO overflow interrupt enabled xx xxxx x1xx = reserved; always read as 0 xx xxxx 1xxx = Host to local mailbox interrupt enabled xx xxx1 xxxx = reserved; always read as 0 xx xx1x xxxx = DMA complete interrupt enabled xx x1xx xxxx = I_2O inbound post FIFO not empty interrupt enabled xx 1xxx xxxx = I_2O outbound post FIFO not empty interrupt enabled x1 xxxx xxxx = PCI target abort interrupt enabled 1x xxxx xxxx = PCI master abort interrupt enabled **Note** All enable bits are initially cleared.
9:0 Interrupt Status	Interrupt Event Status 00 0000 0000 = no events active xx xxxx xxx1 = I_2O Local FIFO overflow xx xxxx xx1x = I_2O PCI FIFO overflow xx xxxx x1xx = reserved; always read as 0 xx xxxx 1xxx = Host to local mailbox xx xxx1 xxxx = reserved; always read as 0 xx xx1x xxxx = DMA operation complete xx x1xx xxxx = I_2O inbound post FIFO not empty (mirror of I2OLISR[3]; read only at this address) xx 1xxx xxxx = I_2O outbound post FIFO not empty (mirror of I2OHISR[3]; read only at this address) x1 xxxx xxxx = PCI target abort 1x xxxx xxxx = PCI master abort **Note** When an event status bit is active, writing a '1' to that bit location clears the bit except for bits 6 and 7. All event status bits are initially cleared.

Local to Host Data Mailbox—LHDATA0x04F8

31 25	24	23 16	15 8	7 0
	I		Byte 1	Byte 0

Bit	Description
24 -- I	Interrupt to Host When this bit is written to 1 by the local processor, it causes a mailbox interrupt to the host. The interrupt remains active until it is cleared by the host in the Host Interrupt Control and Status Register (HINT). 0 = inactive 1 = active This bit is write only.
15:8 -- Byte 1 7:0 -- Byte 0	Data bytes Two bytes of data written by the local and read by the host processor.

Local Bus Configuration Register—LBUSCFG0x04FC

31 21	20 0
	Local Bus Configuration

Bit	Description
20	**LINE_WRAP_DIS:** Defines the enable for cache line wrapping. 0 = Enable cache line wrapping (default)1 = Disable cache line wrapping
19	**RDY_IN_FALL:** Defines the edge of CLKIN used to sample the RDY_IN and RDY_IN input signals. 0 = Rising Edge (default)1 = Falling Edge
18	**SELECT_POL:** Defines the polarity of the SELECT input signal. 0 = Active LOW (default)1 = Active HIGH
17	**SELECT_FALL:** Defines the edge of CLKIN used to sample the SELECT input signal. 0 = Rising Edge (default)1 = Falling Edge
16	**RDY_OUT_OE:** Defines the tri-state mode of the RDY_OUT output signal. 0 = Drive all of the time.1 = Drive only when asserted active. (default)
15	**XTND_RDY_OUT:** Defines the RDY_OUT output signal relation to the final data phase. 0 = Normal. RDY_OUT goes inactive after the final data phase (default) 1 = Extended Ready Out. RDY_OUT remains active after the final data phase until the internal address strobe (typically STROBE) goes inactive. (see field RWMODE for the defining characteristics of the Internal Address Strobe). DO NOT set XTND_RDY_OUT = 1 when BLASTMODE = 1.
14	**BURST_STYLE:** Defines the data ordering protocol of bursts on the local bus. 0 = normal linear bursts (default)1 = 486 style burst (byte ordering in a burst is 048C; 40C8; 8C04; C840)
13	**INT_POL:** Defines the polarity of the IRQ_OUT output signal. 0 = Active LOW interrupt to the local processor (default)1 = Active HIGH interrupt to the local processor
12	**BLAST_POL:** Defines the polarity of the BLAST input signal. 0 = Active LOW (default)1 = Active HIGH
11	**ALE_POL:** Defines the polarity of the ALE input signal. 0 = Active LOW1 = Active HIGH (default)
10	**RDYOUT_POL:** Defines the polarity of the RDY_OUT output signal. 0 = Active LOW (default)1 = Active HIGH
9:8	**BW:** Defines the data bus width of the local processor interface. 00 = 8 bit 01 = 16 bit 10 = 32 bit 11 = 32 bit with encoded byte enables per Motorola protocol (default)

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>7</td><td><b>BLASTMODE:</b> Determines the function of the BLAST input signal.
0 = BLAST is active only during the last transaction of the burst (default)
1 = BLAST is active throughout the entire burst, and goes inactive when with RDY_IN or RDY_IN become inactive on the last read or write of the burst. DO NOT set BLASTMODE = 1 when XTND_RDY_OUT = 1.</td></tr>
<tr><td>6</td><td><b>BEMODE:</b> Determines the byte enable encoding for 16 and 32 bit Motorola modes.
0 = normal byte enables1 = Motorola byte enable encoding. (default)</td></tr>
<tr><td>5:4</td><td><b>RWMODE:</b> Defines how the READ, WRITE, and address STROBE input signals are interpreted internally and defines the Internal Address Strobe. The active polarity of STROBE is determined by ASMODE. ‘01’ is default.
<table>
<tr><th>Pin Name</th><th>RWMODE = 00</th><th>RWMODE = 01</th><th>RWMODE = 1X</th></tr>
<tr><td>READ</td><td>W_R</td><td>Not Used</td><td>READ data; used as Internal Strobe</td></tr>
<tr><td>WRITE</td><td>Not Used</td><td>R_W</td><td>WRITE data; used as Internal Strobe</td></tr>
<tr><td>STROBE</td><td>Internal Address Strobe</td><td>Internal Address Strobe</td><td>Not used as Internal Address Strobe</td></tr>
</table></td></tr>
<tr><td>3:2</td><td><b>ASMODE:</b> Bit 2 defines the polarity of STROBE input signal. And bit 3 defines the edge of CLKIN used to sample the Internal Address Strobe (see field RWMODE for a defining characteristic of the Internal Address Strobe)
x0 = STROBE is active LOW (default)x1 = STROBE is active HIGH
0x = Internal Address Strobe rising edge sampled (default)1x = Internal Address Strobe falling edge sampled</td></tr>
<tr><td>1</td><td><b>DDIN:</b> Delayed Data Input—Defines protocol for validated input data.
0 = input data is valid during the current cycle when RDY_IN, RDY_IN, and RDY_OUT are active. (default)
1 = input data is valid one cycle after when RDY_IN, RDY_IN, and RDY_OUT are active.</td></tr>
<tr><td>0</td><td><b>DDOUT:</b> Delayed Data Output—Defines protocol for validated output data.
0 = output data is valid during current cycle when RDY_IN, RDY_IN, and RDY_OUT are active. (default)
1 = output data is valid one cycle after when RDY_IN, RDY_IN, and RDY_OUT are active.</td></tr>
</table>

Performance Characteristics

Absolute Maximum Ratings [4]

Storage Temperature .... –55°C to +125°C

Ambient Temperature Under Bias .... –40°C to +85°C

Max Operating Current (I_{DD})[5, 6] .. ..250 mA

Voltage on Any V_{DD} Pin Referenced to V_{SS}...–0.5V to +4.0V

Voltage on Any Signal Pin Referenced to V_{SS}–0.5V to +7.0V

Recommended Operating Environment

Ambient Operating Temperature .... T_A0°C to +70°C

Supply Voltage.. V_{DD}+3.0V to +3.6V

Ground Voltage Reference .... V_{SS}0.0V

F_{CLK} (PCI Clock Input Frequency) ......CLK0 MHz to 33 MHz

F_{CLKIN}
(Local Bus Clock Input Frequency)[7] ...CLKIN0 MHz to 50 MHz

Recommended Operating DC Parameters—PCI Bus Signals

The CY7C09449PVA is compatible with the PCI requirements for 3.3V and 5V signaling. Refer to the *PCI Local Bus Specification, Revision 2.2*, as published by the PCI Special Interest Group; the URL is http://www.pcisig.com/

Due to the 5V tolerant nature of the IO, the IO are not clamped to VDD. Operation of the CY7C09449PVA in a PCI 5V signaling environment is electrical and timing compatible with the PCI specification. In a 3.3V signaling environment, all PCI requirements are met except for the output 3.3V clamp, which is in direct conflict with 5V tolerance. The CY7C09449PVA complies with the PCI AC specifications.

Recommended Operating DC Parameters—PCI Bus Signals

The recommended operating DC parameters for the PCI bus are specified here.

Parameter	Description	Condition	Min	Max	Unit
V_{IH}	Input High Voltage		$0.5V_{DD}$	+5.75V	V
V_{IL}	Input Low Voltage		–0.5	$0.3V_{DD}$	V
V_{IPU}	Input Pull up Voltage		$0.7V_{DD}$		V
I_{IL}	Input Leakage Current	$0 < V_{IN} < V_{DD}$		±10	µA
V_{OH}	Output High Voltage[8]	I_{OUT} = –0.5 mA	$0.9V_{DD}$		V
V_{OL}	Output Low Voltage	I_{OUT} = 1.5 mA		$0.1V_{DD}$	V
C_{IN}	Input Pin Capacitance			10	pF
C_{CLK}	CLK Pin Capacitance		5	12	pF
C_{IDSEL}	IDSEL Input Pin Capacitance			8	pF
L_{PIN}	Pin Inductance			20	nH

Recommended Operating DC Parameters—Local Signals

The recommended operating DC parameters for the local bus are specified here.

Parameter	Description	Condition	Min	Max	Unit	Notes
V_{IH}	Input High Voltage		2.0	5.75	V	
V_{IL}	Input Low Voltage		–0.5	0.8	V	
I_{IL}	Input Leakage Current	$0 < V_{IN} < V_{DD}$		±10	µA	
V_{OH}	Output High Voltage	I_{OUT} = –0.8 mA	2.4		V	8
V_{OL}	Output Low Voltage	I_{OUT} = 0.8 mA		0.5	V	
C_{IN}	Input Pin Capacitance			10	pF	
C_{CLK}	CLK Pin Capacitance		5	12	pF	
L_{PIN}	Pin Inductance			20	nH	

Notes

[4] The voltage on any input or IO pin cannot exceed the power pin during power up.

[5] CLK=33 MHz, CLKIN = 50 MHz, PCI and local buses operating at 25% duty cycle.

[6] Also see Operating Power Characteristics on page 47.

[7] For proper initialization, CLKIN must toggle more than 300,000 cycles after $\overline{RST}$ has been deasserted.

[8] Except INTA and IRQ_OUT, which are open drain outputs.

Timing Parameters—PCI Bus Signals

The CY7C09449PVA is compliant with the PCI timing requirements for 3.3V and 5V signaling. Refer to the *PCI Local Bus Specification, Revision 2.2*, as published by the PCI Special Interest Group; the URL is http://www.pcisig.com/.

Parameter	Description	Min	Max	Unit
t_{CYC}	CLK Cycle Time[9]	30		ns
t_{HIGH}	CLK High Time	11		ns
t_{LOW}	CLK Low Time	11		ns
	CLK Slew Rate	1	4	V/ns
t_{VAL}	CLK to Output for Bused Signals[10]	2	11	ns
$t_{VAL(REQ)}$	CLK to Output for REQ[10]	2	12	ns
t_{ON}	Float to Active Delay from CLK	2		ns
t_{OFF}	Active to Float Delay from CLK		28	ns
t_{SU}	Input Setup Time to CLK for Bused Signals	7		ns
$t_{SU(GNT)}$	Input Setup Time to CLK for GNT	10		ns
t_{HOLD}	Input Hold Time to CLK	0		ns
	RST Slew Rate[11]	50		mV/ns
t_{RST}	RST Active Time after Power Stable	1		ms
$t_{RST-CLK}$	RST Active Time after CLK Stable[12]	100		µs
$t_{RST-OFF}$	RST Active to Output Float Delay		40	ns
$t_{RST-FPCA}$	RST High to First PCI Configuration Access	2^{25}		clocks
$t_{RHI-FFA}$	RST High to First FRAME Assertion	5		clocks

Timing Parameters—CY7C09449PVA Buffered PCI Clock and Reset

The CY7C09449PVA provides copies of the PCI clock input, CLK, on the PCLKOUT[2:0] pins. The system level function and timing of these outputs are the same as those of the CLK input. The CY7C09449PVA also provides a registered copy of the PCI reset input, RST, on the RSTOUT pin. The PCI reset is synchronized to the local bus clock, CLKIN. RSTOUT follows RST by no more than two CLKIN cycles. The detailed timing characteristics of the PCLKOUT[2:0] and RSTOUT signal outputs is shown below:

Parameter	Description	Min	Max	Units
$t_{PCLKOUT}$	PCLKOUT Delay from CLK[13]	2	10	ns
t_{RSTOUT}	CLKIN to RSTOUT Valid[13]	2	10	ns

Notes

[9] Clock frequency may range from nominal DC to 33 MHz. The clock frequency may change at anytime, but must not violate other parameters of this specification: clock edges must remain monotonic and within the specified CLK Slew Rate and clock high and low times must be no shorter than specified CLK High and CLK Low Times.

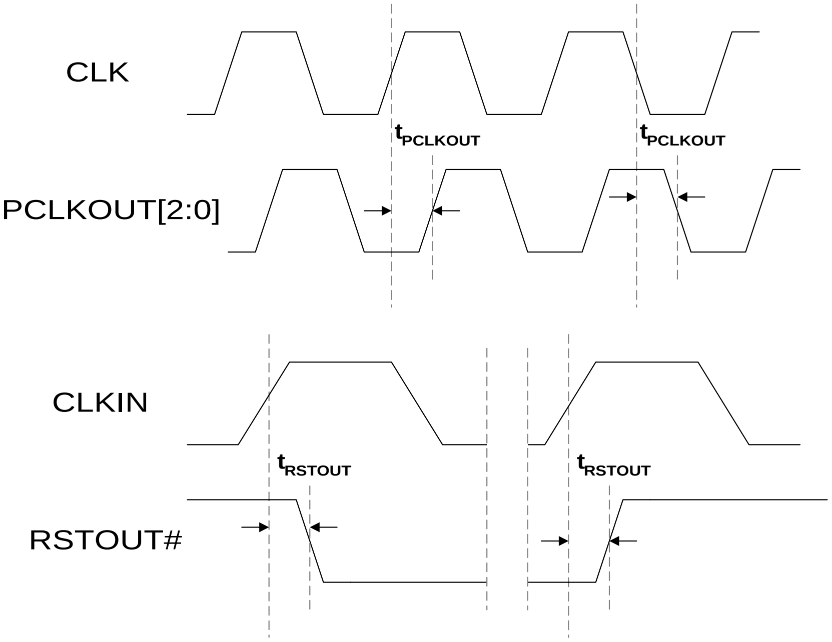
[10] Output maximum times are evaluated with C_L = 50 pF. Output minimum times are evaluated with C_L = 0 pF. Actual test capacitance may vary, but results are correlated to these loads.

[11] Specification only applies to rising (deasserted) edge of RST.

[12] RST is asserted and deasserted asynchronously to CLK.

[13] 50 pF load.

Figure 16. Timing Diagram: PCLKOUT and $\overline{\text{RSTOUT}}$ Signals



Timing Parameters—Local Bus Signals

The parameters for the local bus are specified here.

Parameter	Description	Min	Max	Unit
t_{LOCAL}	CLKIN Cycle Time (Local clock)[14]	20		ns
t_{HIGH}	CLKIN High Time[15]	40	60	%
t_{LOW}	CLKIN Low Time[15]	40	60	%
t_{SU}	Input Setup Time to CLKIN[16]	8		ns
t_{HOLD}	Input Hold Time to CLKIN	3		ns
t_{OUT}	CLKIN to Output Valid[17]	2	10	ns
t_{ON_DQ}	DQ[31:0] Float to Active Delay from CLKIN	2	14	ns
t_{OUT_DQ}	DQ[31:0] Output Delay from CLKIN	2	10	ns
t_{OFF_DQ}	DQ[31:0] Active to Float Delay from CLKIN	2	14	ns
t_{SU_ADR}	ADR[14:2] Input Setup Time to CLKIN	6		
t_{H_ADR}	ADR[14:2] Input Hold Time from CLKIN	3		ns
t_{SU_ALE}	ADR[14:2] Input Setup Time to ALE	1		
t_{H_ALE}	ADR[14:2] Input Hold Time from ALE	3		ns
t_{MIN_ALE}	Minimum Active Pulse width for ALE[18]	5		ns

Notes

[14] V_{TEST} = 1.5V.

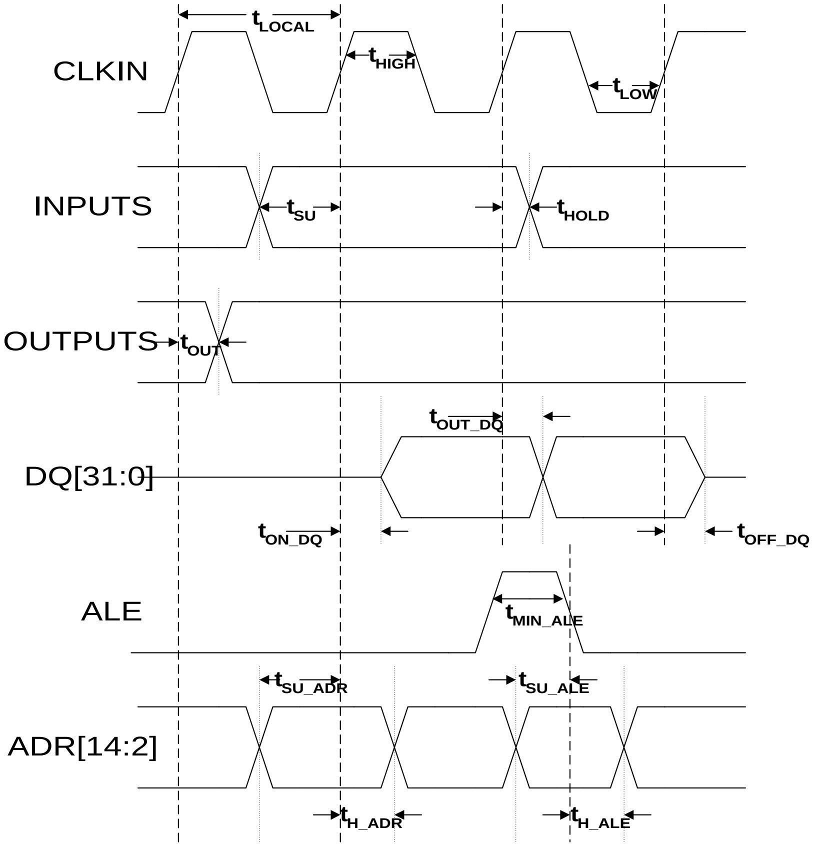
[15] Voltage threshold for HIGH is 2.0V; Voltage threshold for LOW is 0.8V.

[16] Inputs are STROBE, SELECT, READ, WRITE, RDY_IN, RDY_IN, BE[3:0], DQ[31:0], BLAST, and IRQ_IN.

[17] C_L = 50 pF. Outputs are RDY_OUT, IRQ_OUT, RSTOUTD, and RSTOUTD.

[18] Voltage threshold for HIGH is 2.0V.

Figure 17. Timing Diagram: Local Bus Signals



Timing Parameters—I^{2}C Serial Port Bus Signals

The I^{2}C compatible serial interface is designed for a 100 Kb transfer rate. The interface clock is referenced to the local clock, CLKIN. The table below gives the parameters of the CY7C09449PVA's I^{2}C compatible serial interface with respect to the number of local clock periods and the equivalent number of microseconds if the clock is run at 50 MHz. The 100 Kbit/s rate is accomplished with a CLKIN rate of 50 MHz. For CLKIN rates other than 50 MHz, use the Minimum Clocks column to calculate the Minimum Time for each parameter.

Parameter	Description	Minimum Clocks (Clock Periods)	Minimum Time (microseconds)
t_{SCL_LO}	Low Period of SCL	250	5.00
t_{SCL_HI}	High Period of SCL	250	5.00
t_{BUF}	Bus Free Time between 'Start' & 'Stop'[19, 20]	500	10.00
t_{SU_STA}	Setup Time for Repeated 'Start'[19]	250	5.00
t_{HD_STA}	Hold Time for 'Start'	250	5.00
t_{SU_DAT}	Setup Time for Data	125	2.50
t_{HD_DAT}	Hold Time for Data	125	5.00
t_{SU_STO}	Setup time for 'Stop'	250	5.00

Notes

[19] 'Start' condition is a HIGH to LOW transition on SDA while SCL is HIGH.

[20] 'Stop' condition is a LOW to HIGH transition on SDA while SCL is HIGH.

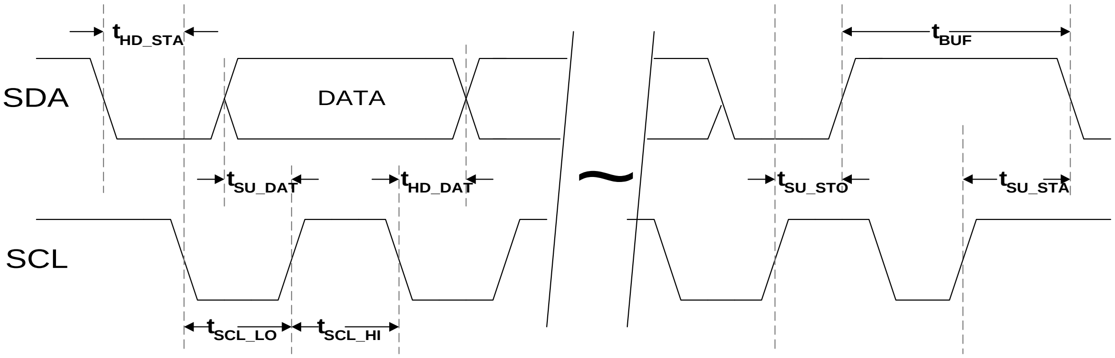


Figure 18. Timing Diagram: I^2C Serial Port Bus Signals

Operating Power Characteristics

Operating power and currents for the CY7C09449PVA at typical environment are specified here, V_{DD} = +3.3V, Temp. = +25°C.

Parameter	Description	Condition	Max	Unit
PD	Power Dissipation[21]		720	mW
I_{DD}	Operating Current[21]		200	mA
$I_{DD\_LSTATIC}$	Static Local Bus Clock[22]		40	mA
I_{DD_STATIC}	Static, no clocks		1	mA

Notes

[21] CLK = 33 MHz, CLKIN = 50 MHz, PCI and local buses operating at 25% duty cycle. This value is typical.

[22] CLK = 33 MHz, CLKIN = 0 MHz, PCI and local buses are inactive. Note that for proper initialization of the CY7C09449PVA, CLKIN must toggle for some number of cycles after RST# is de-asserted. See Recommended Operating Environment on page 43 for the specification of the CLKIN toggle parameter.

CY7C09449PVA Operations

Local Bus Configurations

The CY7C09449PVA interfaces to several processor families. Local bus configurations words for some processors are indicated here. These are not suitable for all applications for a given processor. The specific application's local processor subsystem architecture may impact some parameters of the local bus configuration word. The 21-bit Local Bus Configuration Operations register, LBUSCFG, is written through the I^2C serial interface upon chip initialization to prepare the CY7C09449PVA local bus for the proper interface protocol.

Processor	LBUSCFG value
Motorola Power QUICC MPC860 (default)	0x010B50
Motorola QUICC 68360	0x018B18
Motorola 68040 (default)	0x010B50
Intel i960	0x010A00
Intel i486	0x016A00
Intel 80186	0x012D21
Hitachi SH7708	0x010E11
Hitachi H8/3048	0x010D00
Texas Instruments TMS320LC31	0x010A91

PCI Bus Mastering

Burst transfers between the CY7C09449PVA 16 KB shared memory and the PCI bus system are performed by the direct memory access (DMA) controller. Setup for the DMA controller is accomplished by programming the Operations Registers of the CY7C09449PVA from either the PCI bus interface or the local bus interface. An indication of a completed DMA is available by polling an Operations register or servicing an interrupt. Ownership of the DMA controller by either the PCI or local bus interfaces is arbitrated by software using the Operations Registers.

The address and transfer size registers operate with DWORD resolution. The lower two bits of each of the address and transfer size fields are ignored. Transfers over the PCI bus are DWORD so all four byte enables of the bus are active when transferring data mastered by the CY7C09449PVA. The full 32-bit PCI address space is supported by the DMA controller. The direction of transfer is determined by the 'W' bit in the DMA Control Register. 'W' is the “Write” bit and is with respect to the CY7C09449PVA “writing” to the PCI bus. The basic sequence to setup a DMA is as follows:

1. Enable the Interrupt Mask for the desired interface if an interrupt on DMA completion is required (for example, LINT[21] = 1 enables the interrupt onto the IRQ_OUT pin).
2. Load the address for the beginning of the transfer block of CY7C09449PVA shared memory. This is the DMA Local Base Address Register, DMALBASE.
3. Load the address for the beginning of the transfer block of PCI bus space. This is the DMA Host Base Address Register, DMAHBASE.
4. Load the size of the transfer block. This is the DMA Size Register, DMASIZE.
5. To initiate the DMA, a write to the least significant byte of the DMA Control Register, DMACTL, starts the controller. Writing a '1' to bit 0 transfers data from the CY7C09449PVA shared memory, (pointed to by DMALBASE), to the PCI bus space, (pointed to by DMAHBASE). This causes write bursts on the PCI bus. Writing a '0' to bit 0 transfers the other direction and cause read bursts on the PCI bus. The CY7C09449PVA bus mastering logic uses the most efficient PCI command available for all of its bursts during the transfer.
6. When the DMA is complete, LINT[5] is set. If interrupts are enabled for DMA completions, then an interrupt is generated. If not, LINT[5] is polled.

An additional option of a PCI bus mastered read transfer involves setting the option to perform non prefetchable PCI reads during transfers into CY7C09449PVA shared memory. This option is set in the DMA Control Register with the PFI flag. Also, ownership of the controller is arbitrated in software with assistance of the L and P bits in the DMA Control Register; these are the Local Bus Ownership and PCI Bus Ownership flags, respectively. See CY7C09449PVA Operations for details.

I_2O Message Unit

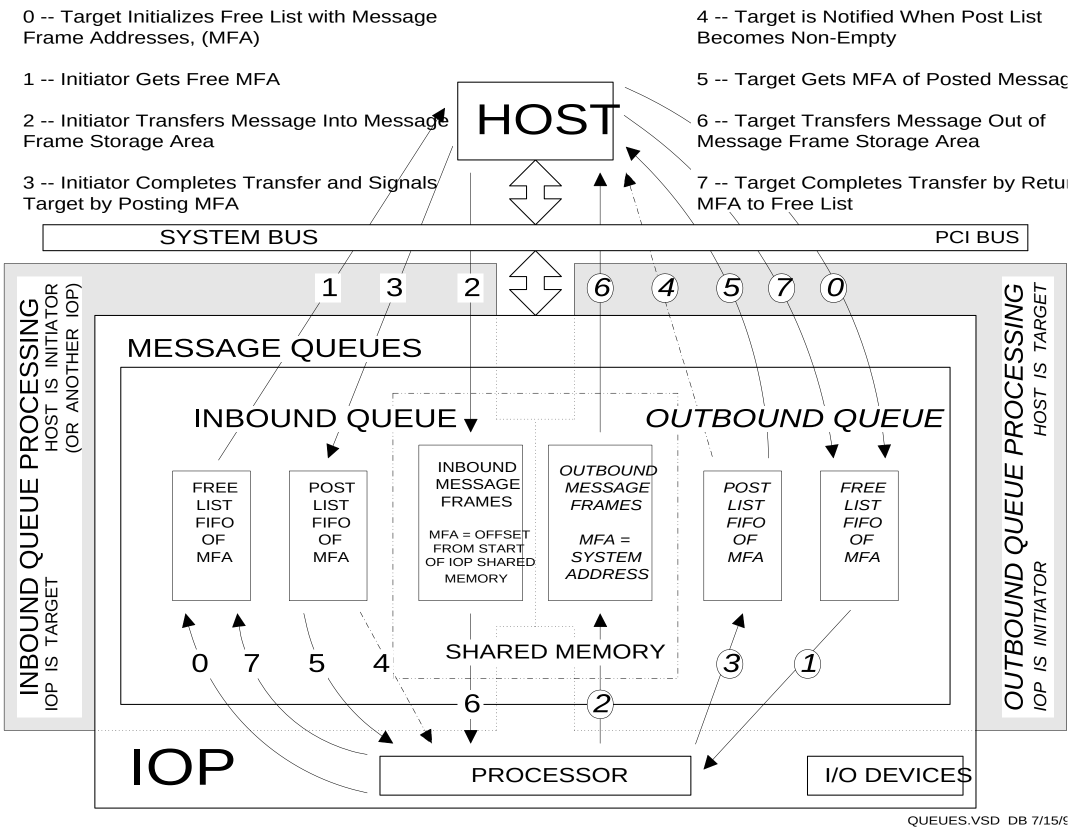
The I_2O specification describes a messaging unit consisting of four FIFOs, a shared memory to store message frames, and an interrupt function. The structure of this unit is described in the I_2O Architecture Specification, version 1.5 on pages 4-2 through 4-7. This capability is fully integrated within the CY7C09449PVA. Reference URL: http://www.i2osig.org/

There is no need for external circuitry to manage the FIFO operations. If I_2O functionality is not required, then the FIFOs are available for general purpose use. Each of the four FIFO are 32 DWORD deep, are accessible from both the PCI and local bus interfaces and generates interrupts to both bus interfaces.

The unit operates in two clock domains, that of the PCI bus and that of the local bus. I_2O message frames for transfer between the PCI and the local domains are located within the 16 Kbyte CY7C09449PVA Shared Memory, which is a general purpose dual-port memory. There is no restriction upon where in the 16 Kbyte space that the message frames reside, however, to satisfy I_2O requirements, the message frames must begin at DWORD boundaries. Neither of the bus's access is dependent upon the operational state of the other bus. This is governed by the nature of the CY7C09449PVA Shared Memory. Operations of I_2O FIFO and the I_2O Interrupt functions occur completely within the clock domain of the local bus. However, access is available to both the PCI and local bus interfaces.

From a system perspective, Figure 19 on page 49 illustrates the I_2O Message Unit transfer function supported by the CY7C09449PVA. The CY7C09449PVA is represented by the “Message Queues” block of the diagram and consists of both Inbound and Outbound Queues and the Shared Memory. For more description of the terminology used in the diagram, refer to the I_2O Architecture Specification. Reference URL: http://www.i2osig.org/.

Figure 19. I$_2$O Message Unit Illustrating Message Queues


0 -- Target Initializes Free List with Message Frame Addresses, (MFA)
1 -- Initiator Gets Free MFA
2 -- Initiator Transfers Message Into Message Frame Storage Area
3 -- Initiator Completes Transfer and Signals Target by Posting MFA
4 -- Target is Notified When Post List Becomes Non-Empty
5 -- Target Gets MFA of Posted Messag
6 -- Target Transfers Message Out of Message Frame Storage Area
7 -- Target Completes Transfer by Retu MFA to Free List
HOST
SYSTEM BUS
PCI BUS
INBOUND QUEUE PROCESSING
HOST IS INITIATOR
(OR ANOTHER IOP)
IOP IS TARGET
OUTBOUND QUEUE PROCESSING
HOST IS TARGET
IOP IS INITIATOR
1
3
2
6
4
5
7
0
MESSAGE QUEUES
INBOUND QUEUE
OUTBOUND QUEUE
FREE LIST FIFO OF MFA
POST LIST FIFO OF MFA
INBOUND MESSAGE FRAMES
MFA = OFFSET FROM START OF IOP SHARED MEMORY
OUTBOUND MESSAGE FRAMES
MFA = SYSTEM ADDRESS
POST LIST FIFO OF MFA
FREE LIST FIFO OF MFA
0
7
5
4
SHARED MEMORY
3
1
6
2
IOP
PROCESSOR
I/O DEVICES
QUEUES.VSD DB 7/15/9

Direct Access

Direct Access allows the local processor to access the PCI bus directly, bypassing the shared memory. In this mode the local processor generates the following PCI bus master cycles:

■ Configuration Read	$C/\overline{BE}[3:0]$ = 0xA
■ Configuration Write	$C/\overline{BE}[3:0]$ = 0xB
■ IO Read	$C/\overline{BE}[3:0]$ = 0x2
■ IO Write	$C/\overline{BE}[3:0]$ = 0x3
■ Memory Read	$C/\overline{BE}[3:0]$ = 0x6
■ Memory Write	$C/\overline{BE}[3:0]$ = 0x7
■ Special Cycle	$C/\overline{BE}[3:0]$ = 0x1
■ Interrupt Acknowledge	$C/\overline{BE}[3:0]$ = 0x0

To operate in this mode, the local processor programs the Direct Access register. Programming sets the base address for the PCI master access and the type of PCI command generated. Then the local processor writes to the Direct Access space of the CY7C09449PVA Memory Map. Offsets into the Direct Access region of the memory map are added to the PCI base address of the Direct Access register and become the address for the PCI bus master access. The type of PCI command generated is defined in the Direct Access register. A local bus read to the Direct Access area of the memory map becomes a PCI bus master read. Similarly, a local bus write to the memory map becomes a PCI bus master write.

Host Bridge

The CY7C09449PVA is used as a host bridge. The processor on the CY7C09449PVA local bus is therefore the host processor in the system. A host processor configures the other PCI devices on the PCI bus. The CY7C09449PVA provides the I^{2}C Serial Port and auto configuration mechanism to setup for host bridge operations. Most aspects of the auto configuration applies to the non host use of CY7C09449PVA.

The CY7C09449PVA must master cycles onto the PCI bus to be a host bridge. The Master Enable bit located in the PCI configuration space is the means to enable CY7C09449PVA PCI mastering. Since the CY7C09449PVA's default value for the Master Enable bit is deasserted, it is necessary to use the CY7C09449PVA auto configuration mechanism to enable PCI mastering. During the power up reset sequence, the I^{2}C serial interface loads data from a non-volatile memory (typically a serial EEPROM) to set the Master Enable bit in the PCI configuration space. Some, but not all, of the PCI configuration values are loaded using this mechanism. These values are read by other devices in the system such as Device ID, Vendor ID, and Class Code to identify the host bridge.

Another part of the auto configuration mechanism is to set up the local bus interface with the host processor and, optionally, provide reset control to the host processor. The Local Bus Configuration register is loaded from the serial EEPROM image. This sets the protocol of the local interface. The Host Control register is loaded by the auto configuration mechanism and controls reset to the host processor. Using the CY7C09449PVA RSTOUTD output signal, (or its complement, RSTOUTD), the CY7C09449PVA controls host processor reset. The Host Control register image is stored in the serial EEPROM and indicates if the reset remains asserted or releases after auto configuration is complete. Normally, it must release the host from reset. If it is not released, an external PCI master is required to release the host processor.

Finally, the CY7C09449PVA uses the Direct Access function to configure PCI devices on the PCI bus. The first device that it configures is typically itself. It is important that the Master Enable bit is set. Without this bit asserted, the CY7C09449PVA cannot configure itself (or any other devices) through the PCI bus.

Dual-Port Shared Memory

To perform concurrent target access to shared memory from the PCI and local bus interfaces it is necessary to devise a handshake protocol and address access allocation scheme to prevent corrupting memory locations. That is, a location within the CY7C09449PVA dual-port memory may be corrupted if a read from one interface occurs simultaneously with a write from the other interface to that same location. The CY7C09449PVA assists the user in managing concurrent access to the shared memory.

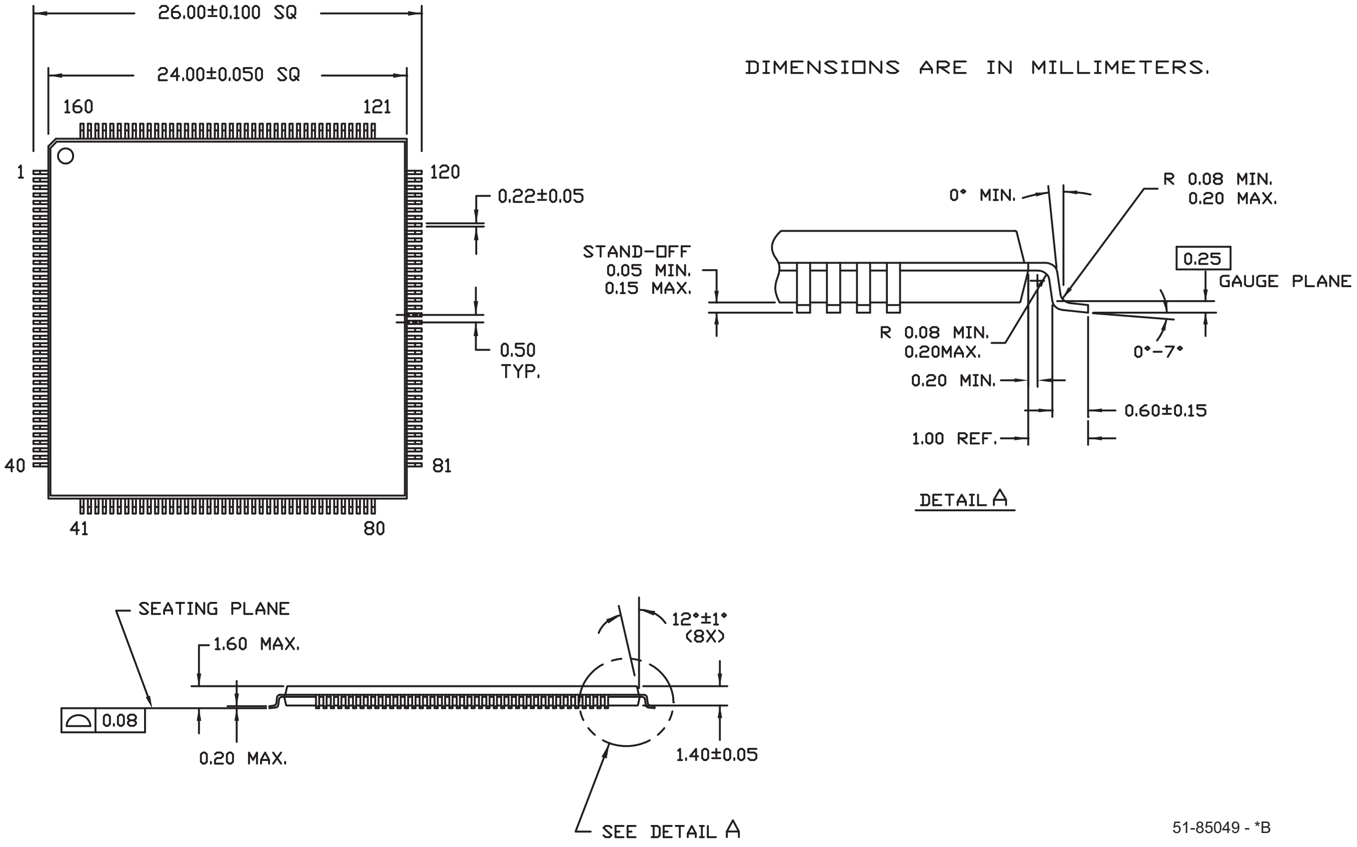
The CY7C09449PVA PCI and local bus are high performance interfaces. Internal logic performs read prefetching to maintain a full speed, zero wait state, burst access to the shared memory. To manage memory access, the CY7C09449PVA performs a disconnect or wait for target reads at each 64 byte boundary. If a user is allocating sections of memory to PCI and local space and intends to execute simultaneous access to the shared memory from both interfaces, then this 64 byte boundary is used to place PCI and local sections of memory adjacent to each other. In other words, the CY7C09449PVA has special logic that detects incoming burst addresses and initiates the disconnect or wait at each 64 byte boundary. In this way, if the transaction is to end at the boundary, then no further prefetching occurs since time is given to the master to end the bus transaction. For the PCI bus, this is performed by a target disconnect. For the local bus, this is wait states.

Ordering Information

Ordering Code	Package Name	Package Type	Operating Range
CY7C09449PVA-AC	51-85049	160-Pin Plastic Thin Quad Flat Pack	0°C to +70°C

Package Diagrams

Figure 20. 160-Pin Plastic Thin Quad Flat Pack



51-85049 - *B

Document History Page

Document Title: CY7C09449PVA-AC 128 Kb Dual-Port SRAM with PCI Bus Controller (PCI-DP) Document Number: 001-40319				
Rev.	**ECN No.**	**Orig. of Change**	**Submission Date**	**Description of Change**
**	2077346	VKN/PYRS	See ECN	New data sheet
*A	2722497	RAME	06/23/09	Minor ECN to post the data sheet to the web
*B	2732450	RAME	07/07/09	Converted from Preliminary to Final

Sales, Solutions, and Legal Information

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at cypress.com/sales.

Products

PSoC	psoc.cypress.com
Clocks & Buffers	clocks.cypress.com
Wireless	wireless.cypress.com
Memories	memory.cypress.com
Image Sensors	image.cypress.com

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.